



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

---

# Scramikub

---

## Memòria

*Autor:*

Oriol BURGAYA VERDAGUER

Grau en Enginyeria Informàtica

Especialitat en Enginyeria del Software

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya

*Director:*

Enric MAYOL SARROCA

Departament d'Enginyeria de Serveis i Sistemes d'Informació

Universitat Politècnica de Catalunya

Gener de 2017



## Resum

Scramikub és un joc de taula que combina dos jocs d'allò més coneguts, l'Scrabble i el Rummikub. En aquest projecte s'adaptarà aquest joc de taula a una aplicació mòbil online i multiplataforma, permetent així, jugar amb usuaris de diferents sistemes operatius mòbils d'arreu del món.

Per a aconseguir aquest objectiu, s'utilitzarà un framework que ens permet compilar una aplicació desenvolupada amb tecnologies web (HTML, CSS i Javascript) per a iOS i Android, utilitzant sempre el mateix codi.

Tot i la facilitat a l'hora d'obtenir les versions per a cada sistema operatiu mòbil, caldrà tenir molta cura de tots els controls i comportaments que tingui l'aplicació final, ja que no poden allunyar-se molt dels que utilitzen les aplicacions mòbils natives.

El projecte serà gestionat amb una eina especialitzada (JIRA), seguint una metodologia àgil (SCRUM).

## Abstract

Scramikub is a board game that combines two well-known games, Scrabble and Rummikub. In this project I'll adapt this board game in a multiplatform online application for mobile devices allowing us to play with users of different mobile operating systems around the world.

To achieve this goal, we will use a framework that allows us to compile an application developed with web technologies (HTML, CSS and JavaScript) for iOS and Android, always using the same code.

Despite the ease of obtaining the versions for each mobile operating system, we'll have to take care with all the controls and behaviors that the final application has, since they can not be too different from those used in native mobile applications.

The project will be managed with a specialized tool (JIRA), using an agile methodology (SCRUM).

## Resumen

Scramikub es un juego de mesa que combina dos juegos muy conocidos, el Scrabble y el Rummikub. En este proyecto se adaptará este juego de mesa en una aplicación móvil online y multiplataforma, permitiendo así, jugar con usuarios de diferentes sistemas operativos móviles de todo el mundo.

Para conseguir este objetivo, se utilizará un framework que nos permitirá compilar una aplicación desarrollada con tecnologías web (HTML, CSS y JavaScript) para iOS y Android, utilizando siempre el mismo código.

A pesar de la facilidad a la hora de obtener las versiones para cada sistema operativo móvil, habrá que tener mucho cuidado con todos los controles y comportamientos que tenga la aplicación final, ya que no pueden alejarse mucho de los que utilizan las aplicaciones móviles nativas.

El proyecto será gestionado con una herramienta especializada (JIRA), siguiendo una metodología ágil (SCRUM).

# Índex

<b>1</b>	<b>Introducció o Context</b>	<b>8</b>
1.1	Introducció . . . . .	8
1.2	Motivació . . . . .	8
1.3	Reglament . . . . .	9
1.3.1	Inventari . . . . .	9
1.3.2	Mecànica de joc . . . . .	9
1.3.3	Combinacions . . . . .	11
1.4	Actors implicats . . . . .	15
<b>2</b>	<b>Estat de l'Art</b>	<b>16</b>
2.1	Precedents . . . . .	16
2.1.1	Words With Friends (Apalabrados) . . . . .	17
2.1.2	Alphabear . . . . .	18
2.1.3	Yushino . . . . .	19
2.2	Tecnologies candidates . . . . .	19
2.2.1	Unity . . . . .	19
2.2.2	Xamarin . . . . .	20
2.3	Tecnologia escollida . . . . .	20
2.3.1	Part client (Ionic) . . . . .	20
2.3.2	Part servidor (Firebase) . . . . .	20
2.3.3	Entorn de desenvolupament . . . . .	20
<b>3</b>	<b>Metodologia</b>	<b>21</b>
3.1	Metodologia de treball . . . . .	21
3.2	SCRUM . . . . .	21
3.3	Eina de seguiment (JIRA) . . . . .	21
<b>4</b>	<b>Abast</b>	<b>24</b>
4.1	Definició de l'abast . . . . .	24
4.2	Possibles obstacles . . . . .	24
4.2.1	Tecnologia nova . . . . .	24
4.2.2	Multiplataforma . . . . .	24
4.2.3	Arquitectura mal dissenyada . . . . .	25
<b>5</b>	<b>Objectius del projecte</b>	<b>26</b>
5.1	Objectius . . . . .	26
5.2	Competències tècniques . . . . .	27

<b>6</b>	<b>Planificació temporal</b>	<b>28</b>
6.1	Planificació temporal . . . . .	28
6.1.1	Planificació estimada del projecte . . . . .	28
6.1.2	Recursos . . . . .	28
6.1.3	Pla d'acció i valoració d'alternatives . . . . .	28
6.2	Descripció de les tasques . . . . .	29
6.2.1	Estudi de la tecnologia escollida . . . . .	29
6.2.2	Preparació de l'entorn . . . . .	29
6.2.3	Implementació de la gestió usuaris . . . . .	29
6.2.4	Implementació del joc . . . . .	29
6.2.5	Retocs finals . . . . .	30
6.2.6	Primera Release de l'aplicació i entrega de la documentació .	30
6.2.7	Desviacions en la planificació . . . . .	30
<b>7</b>	<b>Gestió econòmica</b>	<b>31</b>
7.1	Identificació dels costos . . . . .	31
7.2	Estimació dels costos . . . . .	31
7.2.1	Costos directes per tasques . . . . .	31
7.2.2	Costos indirectes . . . . .	32
7.2.3	Pressupost final . . . . .	33
7.3	Control de gestió . . . . .	33
<b>8</b>	<b>Sostenibilitat</b>	<b>34</b>
8.1	Econòmica . . . . .	34
8.2	Social . . . . .	34
8.3	Ambiental . . . . .	34
<b>9</b>	<b>Especificació</b>	<b>35</b>
9.1	Requisits . . . . .	35
9.1.1	Requisits d'interfície . . . . .	35
9.1.2	Requisits funcionals . . . . .	36
9.1.3	Requisits no funcionals . . . . .	36
9.2	Casos d'ús de l'aplicació mòbil . . . . .	37
9.2.1	Casos d'ús relacionats amb l'autenticació d'usuari . . . . .	37
9.2.2	Casos d'ús Lobby . . . . .	37
9.2.3	Casos d'ús Settings . . . . .	37
9.2.4	Casos d'ús Ranking . . . . .	37
9.2.5	Casos d'ús User . . . . .	37
9.2.6	Casos d'ús Creació de partida . . . . .	38
9.2.7	Casos d'ús Joc . . . . .	38
9.2.8	Casos d'ús Xat . . . . .	39

9.3	Casos d'ús de l'Administrador Web . . . . .	39
9.4	Model Conceptual del Sistema . . . . .	40
<b>10</b>	<b>Disseny del Sistema</b>	<b>41</b>
10.1	Arquitectura Global del Sistema . . . . .	41
10.1.1	Patró de disseny : MVC . . . . .	41
10.2	Mapa navegacional . . . . .	43
10.3	Tecnologia i eines usades pel desenvolupament . . . . .	45
10.3.1	Ionic Framework . . . . .	45
10.3.2	Firebase . . . . .	47
10.4	Interfície gràfica . . . . .	48
10.4.1	Login i Registre . . . . .	48
10.4.2	Lobby . . . . .	49
10.4.3	Informació d'usuari . . . . .	50
10.4.4	Creació de partida . . . . .	51
10.4.5	Taulell de joc . . . . .	53
10.4.6	Xat . . . . .	55
10.5	Integració en varis Sistemes Operatius . . . . .	56
10.6	Usabilitat . . . . .	56
10.7	Disseny de la Base de Dades . . . . .	59
10.7.1	Users . . . . .	59
10.7.2	Games . . . . .	60
10.7.3	Chats . . . . .	61
<b>11</b>	<b>Desenvolupament</b>	<b>62</b>
11.1	Funcionalitats . . . . .	62
11.1.1	Funcions d'administració d'usuaris . . . . .	62
11.1.2	Funcions d'administració de partides . . . . .	64
11.1.3	Implementació del procés de joc . . . . .	65
11.1.4	Altres funcionalitats . . . . .	70
11.1.5	Administrador extern de l'aplicació . . . . .	71
<b>12</b>	<b>Adaptacions realitzades</b>	<b>75</b>
12.1	Adaptacions del reglament . . . . .	75
12.2	Adaptacions gràfiques . . . . .	75
<b>13</b>	<b>Problemes trobats durant el desenvolupament</b>	<b>76</b>
13.1	CrossPlatform . . . . .	76
13.2	Native gestures . . . . .	76
13.3	Rendiment . . . . .	76

<b>14 Conclusions i Treball Futur</b>	<b>78</b>
<b>Referències</b>	<b>80</b>



# 1 Introducció o Context

## 1.1 Introducció

**Scramikub** és un joc de taula ideat i dissenyat per Àlex González Rodríguez, conjuntament amb Enric Mayol Sarroca, professor de la Facultat d'Informàtica de Barcelona, i que neix per hibridació de dos jocs prou coneguts: **Scrabble** i **Rummikub**. El joc pretén mantenir la mecànica de Rummikub però limitat per un tauler on es permet el creuament de sèries numèriques a imatge de com ho fan els mots a Scrabble.

En aquest treball es recrearà i adaptarà Scramikub en una aplicació o videojoc per a dispositius mòbils. Aquesta, serà totalment online i cross-platform, és a dir, un usuari podrà jugar amb qualsevol altre usuari de l'aplicació, i no serà necessari que els dos tinguin el mateix sistema operatiu mòbil. És a dir, un usuari del sistema operatiu **Android** pot jugar amb un altre que tingui un dispositiu amb **iOS**.

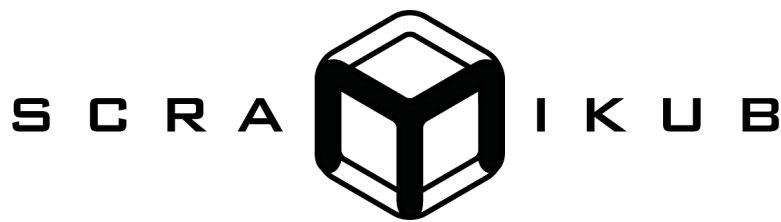


Figura 1: Logo provisional del joc

## 1.2 Motivació

La motivació per a tirar endavant sorgeix de la passió pels jocs de taula que tinc desde ben jove, sobretot en un dels jocs amb el que s'inspira el joc que s'adaptarà en aquest projecte, el *Rummikub*.

A part d'això, desde que vaig començar en el món del desenvolupament de software sempre he tingut certes ganes de desenvolupar per a dispositius mòbils, encara que em dediqui a altres tipus de software professionalment.

Tot això va fer que em poses en contacte amb el meu director del projecte, Enric Mayol, quan vaig veure la seva oferta en el Racó i decidís tirar endavant aquest projecte amb molta il·lusió i ganes.

## 1.3 Reglament

Per a entendre millor el joc de taula que s'adaptarà per a dispositius mòbils, es redactarà a continuació un petit reglament d'aquest.

### 1.3.1 Inventari

Aquí definirem tot el material que necessitarem per a realitzar una partida de Scramikub.

- Taulell de 15 per 15.
- 30 boles blaves en dues sèries numerades de l'1 al 15.
- 30 boles vermelles en dues sèries numerades de l'1 al 15.
- 30 boles grogues en dues sèries numerades de l'1 al 15.
- 30 boles liles en dues sèries numerades de l'1 al 15.
- 30 boles verdes en dues sèries numerades de l'1 al 15.
- 30 boles taronges en dues sèries numerades de l'1 al 15.
- 6 boles especials que seràn els comodins.
- Rellotge de sorra de 1, 2 i 3 minuts.

El nombre de boles de colors i especials vindrà donada pel número de jugadors, tot i això, aquests són lliures de decidir quants pals i comodins volen utilitzar durant la partida sense cap mena de restricció.

### 1.3.2 Mecànica de joc

Tal i com s'ha dit anteriorment, abans de començar la partida, es definirà amb quants pals de colors i comodins es jugarà.

Com a regla general, es jugarà amb tants pals i tants comodins com jugadors hi hagi, amb un mínim de 4 pals i 4 comodins.

Seguidament, es repartiran aleatòriament 15 boles a cada jugador per a començar la partida. Per decidir qui comença a jugar, es repartirà una última bola a cada jugador. El que tingui el número més alt, serà l'encarregat d'encetar el taulell.

En la primera tirada de cada jugador a la partida, caldrà que aquest jugui una combinació que sumi un mínim de 15 punts. A partir de llavors, els jugadors ja podran fer totes les sèries que vulguin (independentment dels punts que sumi).

En el cas de que un jugador no pugui tirar durant el seu torn, aquet podrà agafar una altra bola i intentar fer una jugada amb aquesta en el mateix torn. Si no pot fer cap jugada, haurà de passar el torn al següent jugador.

Les combinacions o sèries que es poden col·locar sobre el taulell hauràn de ser de com a mínim 3 boles o fitxes. Aquestes sèries poden ser de dos tipus:

- Escales de números consecutius del mateix color (tenen continuïtat, és a dir, darrera del número 15 ve l'1). Aquestes escales també es poden formar en sentit contrari.
- Peces del mateix número però de diferents pals o colors (mai repetits).

Amb el taulell de joc passa el mateix que amb les escales: després de la posició B15 ve la B1, per tant podrem fer sèries utilitzant els límits del tauler.

El comodí podrà substituir qualsevol peça, però sempre tindrà el mateix valor. És a dir, si el comodí forma part de dos sèries que es creuen, el valor d'aquest comodí ha de ser el mateix en ambdues.

El sistema de puntuació es pot pactar abans de començar la partida.

Els dos sistemes de puntuació existents fins ara són:

- Guanya el jugador que es queda abans sense peces, és a dir, el primer que les col·loqui totes al taulell de forma correcta.
- Guanya el jugador que hagi col·locat sèries amb més valor al taulell.

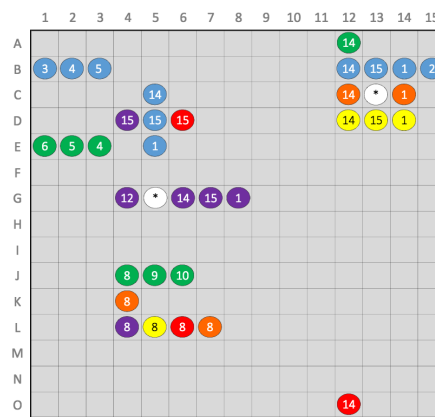


Figura 2: Exemple de partida

### 1.3.3 Combinacions

Per tal d'entendre millor el funcionament del joc *Scramikub*, es portarà a terme una partida entre dos jugadors utilitzant una versió gairebé final de l'aplicació que s'ha desenvolupat durant aquest projecte. Així podrem veure la majoria de combinacions que els jugadors poden fer durant una partida normal.

#### Combinacions possibles

Aquí recollirem varies combinacions correctes que es donen a la partida en procés. Les dos combinacions bàsiques del joc es poden veure a la següent captura.



Figura 3: Exemple de partida

En aquesta figura (Figura 3), podem veure com s'han enllaçat dues combinacions diferents, una sèrie vertical creixent del mateix pal, i una sèrie horitzontal del mateix valor i diferents pals. Com es pot observar, comparteixen la fitxa del comodí i, com no podria ser d'una altra manera, aquesta té el valor d'un 15 de color lila.

En el cas de la combinació vertical de la Figura 3, veiem que la sèrie va de petit a gran, encara que està permès fer sèries a l'inrevés, és a dir, de gran a petit.

Això és el que podem veure a les dos captures següents, on hi ha una combinació horitzontal amb una sèrie de gran a petit (de esquerra a dreta) i una altra vertical amb una sèrie de gran a petit (de dalt a baix).



Figura 4: Combinació horitzontal a l'inrevés



Figura 5: Combinació vertical a l'inrevés

Qualsevol combinació es pot fer utilitzant els costats del taulell, fent que aquesta segueixi pel costat oposat.

Un exemple senzill d'això es pot veure a la següent captura, on hi ha una sèrie horitzontal que comença a la part dreta del taulell i acaba al costat oposat. Aquesta tirada seria totalment correcta.



Figura 6: Combinació utilitzant els límits del taulell

## Combinacions incorrectes

En les següents captures es mostraran dues combinacions errònies que es podrien donar en una partida.



Figura 7: Combinació incorrecte



Figura 8: Combinació incorrecte

En el cas de la Figura 7, la combinació horitzontal és correcta. En canvi, la combinació en vertical és incorrecta ja que només consta de dues fitxes i el mínim de fitxes per a què una combinació sigui correcta és 3.

En l'altra captura és més difícil veure on es troba la combinació errònia, però si ens hi fixem, la combinació de 14's horitzontal només consta de 2 fitxes. La resta de combinacions de la captura són correctes.

## Resultat final

Una vegada finalitzada la partida, el taulell acostuma a quedar més o menys ple de combinacions, depenent de les fitxes dels jugadors, però també de l'experiència d'aquests.

Seguidament es pot veure com ha quedat el taulell de la partida anterior:



Figura 9: Exemple de partida

## 1.4 Actors implicats

- **Creador del joc de taula:** És el dissenyador del joc i el que va idear les regles. Amb aquest, s'haurà d'acordar les adaptacions i canvis que sufreixi el joc, per tal de millorar el funcionament i dinàmica d'aquest en dispositius mòbils.
- **Jugadors o usuaris:** Seran els que utilitzaran l'aplicació i dels quals haurem de tenir molt en compte el seu feedback, ja que serà molt important per a millorar el resultat final del joc.  
D'aquesta manera, obtindrem una aplicació totalment acabada i polida.
- **Oriol Burgaya (desenvolupador):** Serà l'encarregat d'adaptar aquest joc de taula a una aplicació per a dispositius mòbils online i cross-platform.
- **Mercats d'aplicacions:** Són els mercats digitals on es penjaràn les aplicacions desenvolupades. L'aplicació haurà de complir algunes condicions per a ser acceptada en aquests mercats, sobretot en el cas de la versió iOS, on aquestes són molt més estrictes que en el cas del sistema operatiu mòbil Android.



## 2 Estat de l'Art

### 2.1 Precedents

Actualment, el mercat d'aplicacions mòbils disposa d'un ampli ventall d'adaptacions de jocs de taula online. Fins i tot, moltes empreses especialitzades en jocs de taula adapten elles mateixes les seves creacions, deixant de banda el joc físic i connectant milers de jugadors per a què juguin entre ells. Un exemple d'aquestes empreses podria ser Hasbro, que ha adaptat gairebé tots els jocs de taula que ha creat: Monopoly, Risk, Trivial.

Per a dur a terme aquest projecte, és molt important que primer de tot s'analitzin les aplicacions similars que ja estan al mercat en aquest moment. D'aquestes aplicacions caldrà estudiar les seves capacitats i carències, les diferències amb la nostra aplicació, la interfície gràfica i altres aspectes importants a tenir en compte. D'aquesta manera, podrem diferenciar-nos d'aquestes aplicacions, però també agafar idees de les característiques que han fet que tants usuaris juguin amb elles dia rere dia.

Les aplicacions que he estudiat han estat les següents:

### 2.1.1 Words With Friends (Apalabrados)

Words with friends és segurament la millor adaptació a aplicació mòvil del joc de taula **Scrabble** i és un clar exemple d'aplicació cross-platform. Va ser desenvolupat per l'empresa Zynga i disposa de característiques que milloren molt l'experiència d'usuari, com un xat entre els jugadors, notificacions, Smart Match (jugar contra algú aleatori d'arreu del món), entre d'altres. Aquest joc està disponible per a gairebé totes les plataformes disponibles: Android, Windows Phone, iOS. A més a més, els jugadors també poden jugar desde Facebook.



Figura 10: Captura de Words With Friends

### 2.1.2 Alphabear

Alphabear és un joc de puzzle amb paraules on l'objectiu és formar paraules per a aconseguir la major quantitat de punts possibles. A diferència de les altres aplicacions presentades, Alphabear no és multijugador però convé fixar-se en la seva interessant interfície que pot influir molt en una bona experiència d'usuari.



Figura 11: Captura de Alphabear

### 2.1.3 Yushino

Yushino es defineix com a un "mots encreuats" de números, i serà el que, centrant-nos en la interfície gràfica i el seu funcionament, tindrà més similituds amb l'aplicació que es vol desenvolupar. A Yushino, els usuaris hauràn de formar seqüències d'almenys 3 números, on el últim haurà de ser la suma dels dos primers. A diferència de Words with Friends, Yushino no és un joc cross-platform i només està disponible per a iOS. Per tant, es queda una mica enrere dels seus competidors.

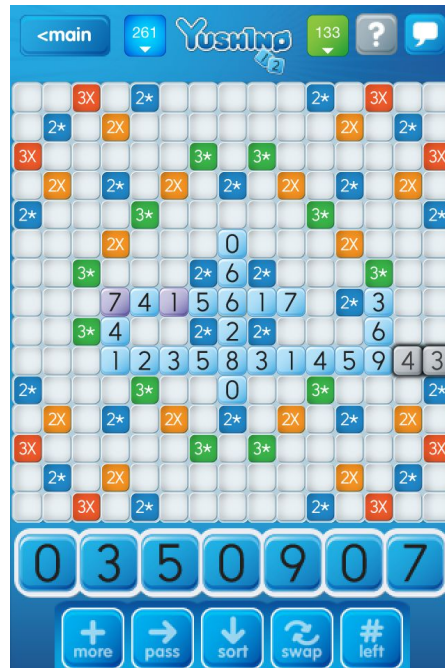


Figura 12: Captura de Yushino

## 2.2 Tecnologies candidates

Ara s'exposaran diferents tecnologies que han estat candidates a ser utilitzades per a portar a terme aquest projecte, però que s'han descartat per vèries raons.

### 2.2.1 Unity

**Unity** és un motor de videojocs multiplataforma, el qual la seva plataforma de desenvolupament té suport de compilació per a diferents tipus de plataformes, entre elles, Android i iOS.

Encara que aquesta opció ens ofereixi suport de compilació pels diferents sistemes operatius mòbils, l'aplicació que volem adaptar no necessita un motor gràfic per

a funcionar i, a més a més, el desconeixament d'aquesta eina fa que la descartem com a tecnologia per a desenvolupar la nostra aplicació.

### 2.2.2 Xamarin

**Xamarin** és una eina que implementa la plataforma de desenvolupament o framework **.NET** per a dispositiu Android i iOS.

Tot i que és una aposta prou forta, queda descartada per el resultat final que tenen les solucions desenvolupades amb aquest framework, ja que no tenen total semblança amb les aplicacions natives de cada sistema operatiu mòbil que tractarem en aquest projecte.

## 2.3 Tecnologia escollida

Després d'estudiar varies tecnologies possibles per a realitzar aquest treball, s'ha arribat a una conclusió bastant clara i prou justificada sobre quina tecnologia escollir.

Aquestes tecnologies, es detallaràn amb molta més precisió a l'apartat de Disseny del Sistema.

### 2.3.1 Part client (Ionic)

**Ionic** és un potentíssim SDK HTML5 que ens facilita una manera de compilar aplicacions que semblen gairebé natives utilitzant tecnologies web com HTML, CSS, i Javascript.

Ionic treballa amb AngularJS, un framework que ens permet exprimir tot el potencial del llenguatge Javascript i utilitzar qualsevol patró de disseny que s'ens acudeixi.

### 2.3.2 Part servidor (Firebase)

**Firebase** és un proveïdor de serveis cloud i, tot i que no utilitzarem tots aquests serveis, alguns ens seràn molt útils (Auth i Storage).

### 2.3.3 Entorn de desenvolupament

Com que la tecnologia per a desenvolupar el front-end de l'aplicació és una tecnologia web, com a entorn de desenvolupament ens servirà amb un editor de text.

En el meu cas, he utilitzat el editor de text **Visual Studio Code**, que ens permet una integració molt completa amb *Git* i disposa de molts plugins per agilitzar el desenvolupament el màxim possible.

## 3 Metodologia

### 3.1 Metodologia de treball

Aquest projecte està planejat per a treballar de forma incremental i en sprints més aviat curts (com a molt 2 setmanes). Una vegada implementada la part imprescindible per a portar a terme partides entre usuaris, seguirem planejant sprints per a arreglar els problemes trobats fins al moment i afegir les noves features fins a tenir el producte totalment pol·lit.

A part d'això, es definiran diferents releases de l'aplicació per tal de tenir diferents versions d'aquesta i aconseguir així un cert control sobre el funcionament de les noves features i com va evolucionant el sistema.

### 3.2 SCRUM

S'utilitzarà SCRUM com a marc de treball per a la gestió i desenvolupament de l'aplicació basada en un procés iteratiu i incremental.

### 3.3 Eina de seguiment (JIRA)

Per a la gestió del projecte, seguiment d'errors i incidències utilitzarem la eina **JIRA**, que és de les més exteses per a aquest propòsit.



Figura 13: Logo JIRA

Amb aquesta eina, podrem planificar els sprints d'una manera fàcil i molt ràpida, minimitzant així tot el temps que invertim en la planificació de tasques.

Aquesta captura és un exemple de sprint que s'ha fet durant el desenvolupament de l'aplicació:

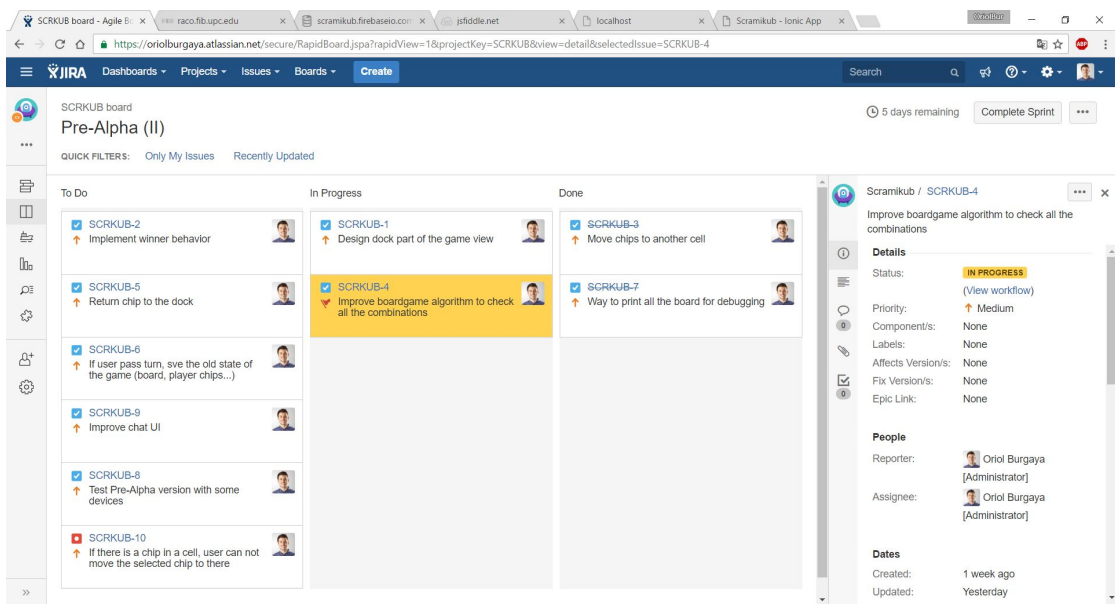


Figura 14: Exemple de sprint a JIRA

Com es pot observar, JIRA ens ofereix una dashboard molt clara on s'ens llisten totes les *issues* assignades en l'sprint actual. Aquestes estan classificades per *To Do*, *In Progress* i *Done*. Això ens permet saber en tot moment les issues que estan en desenvolupament, les que queden per desenvolupar, i les que ja estan acabades. A part d'això, he intentat fer un altre pas de test i verificació de les issues una vegada estaven acabades.

El Backlog que ens ofereix JIRA, ens ha servit per tal de tenir una visió global de totes les issues dels sprints que s'han creat.

En aquesta pantalla, es poden reordenar les issues per prioritat i canviar-les de sprint fàcilment, entre altres funcionalitats molt útils.

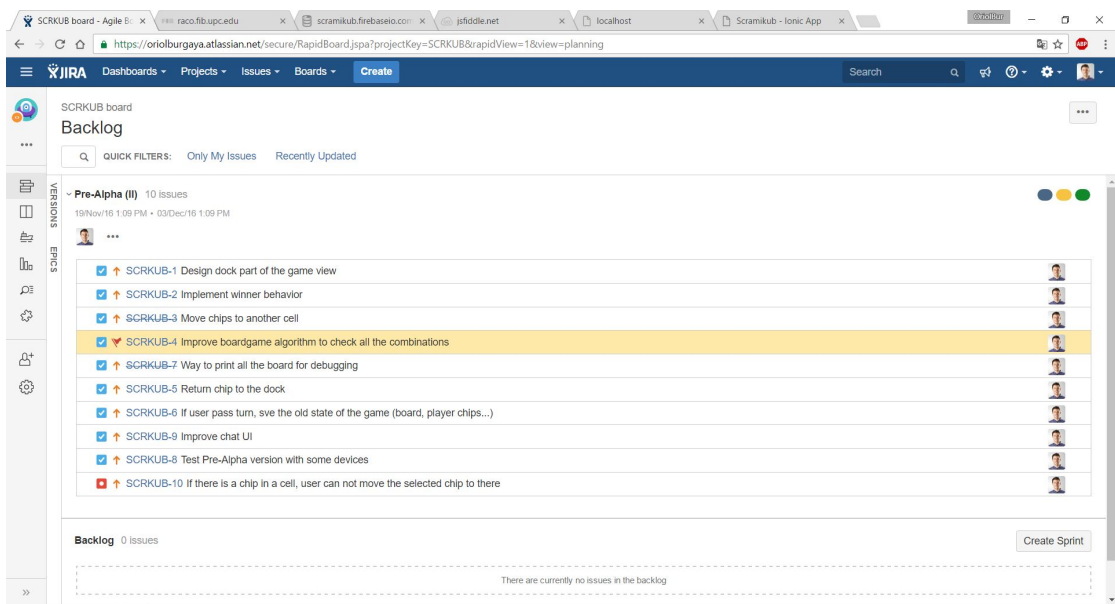


Figura 15: Backlog de JIRA

JIRA també ens ha permès fer un control de les diferents versions de l'aplicació que s'han anat desenvolupant.

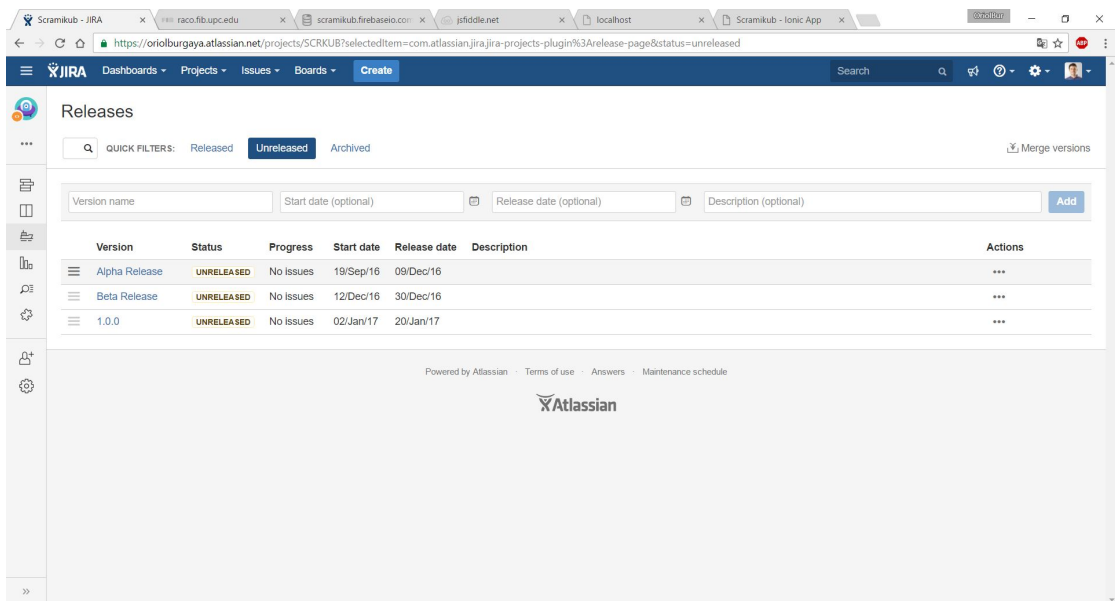


Figura 16: Pantalla de Releases de JIRA



## 4 Abast

### 4.1 Definició de l'abast

L'objectiu serà fer una Release pública de l'aplicació per als sistemes operatius mòbils iOS i Android durant el mes de gener del 2017.

A partir d'això, s'utilitzarà tot el feedback que els usuaris ens donin per tal de resoldre bugs i millorar l'experiència d'aquests.

Aquesta primera versió no tindrà característiques que no siguin imprescindibles per a fer partides amb altres usuaris, és a dir, les característiques com el xat i l'Smart Match entre d'altres potser no estan implementades en aquesta versió.

### 4.2 Possibles obstacles

Durant el disseny i desenvolupament d'aquest projecte, ens trobarem diferents obstacles més o menys importants que segurament ens faràn variar la planificació final d'aquest. Tot i això, si estudiem aquests possibles obstacles serà més fàcil evitar-los i disminuir el seu impacte en la planificació actual.

#### 4.2.1 Tecnologia nova

La tecnologia escollida que utilitzarem per a desenvolupar l'aplicació (Ionic Framework) és relativament nova i encara no està lliure de bugs. Això ens implicarà algunes hores extra de desenvolupament per tal de detectar errors en la tecnologia i tractar d'evitar-los utilitzant algun *work around* que ens permeti arribar al nostre objectiu satisfactòriament.

A més a més, algunes parts de l'aplicació que es durà a terme no són gens trivials i no hi ha molta documentació disponible sobre elles, per exemple, la funcionalitat de drag and drop per a col·locar les peces al tauler.

#### 4.2.2 Multiplataforma

Aquest potser és l'obstacle amb més importància d'aquest projecte, ja que molta part dels esforços invertits en el desenvolupament serà el correcte funcionament del joc en ambdós sistemes operatius mòbils, Android i iOS.

Encara que Ionic ens ajudarà a aconseguir aquest objectiu, hi hauràn certes parts de l'aplicació en què ens trobarem diferències entre les dos plataformes, com per exemple, els recursos que utilitzi l'aplicació (imatges, icones...), la disposició d'alguns controls...

A part d'això, cal dir que el nostre joc no és una aplicació estàndard que es podria desenvolupar fàcilment amb aquest framework.

### **4.2.3 Arquitectura mal dissenyada**

Un aspecte molt important del projecte serà la realització d'un molt bon disseny d'arquitectura per tal de que, una vegada acabat aquest, l'aplicació sigui totalment escalable en quan a quantitat de jugadors actius en el sistema, i que sigui relativament fàcil arreglar qualsevol mena de bug i afegir noves features.

Per tant, caldrà escollir el patró que més s'adapti a l'aplicació que es desenvoluparà per fer que sigui fàcilment mantenible en un futur pròxim.

## 5 Objectius del projecte

### 5.1 Objectius

En aquest projecte s'intentaràn assolir diferents objectius bastant marcats:

- **Desenvolupament d'una aplicació que adapti el joc Scramikub a dispositius mòbils:** Aquest serà l'objectiu principal d'aquest projecte, el desenvolupament de l'aplicació. Aquesta haurà de ser totalment funcional en qualsevol sistema operatiu mòbil suportat (Android i iOS), i permetrà que dos usuaris de sistemes operatius diferents puguin interactuar entre ells. Al assolir aquest objectiu, tindrem una primera release pública de l'aplicació per als dos sistemes operatius mòbils.
- **Tria de la tecnologia que més s'adapti al projecte actual:** Serà necessari fer un estudi previ de les diferents tecnologies que ens permetin desenvolupar l'aplicació satisfactòriament i de la manera més eficaç. En el cas d'aquest projecte, necessitarem una tecnologia viable per a desenvolupar una aplicació cross-platform online per a dispositius mòbils.
- **Ús d'una metodologia àgil de treball:** S'utilitzarà una metodologia àgil per a desenvolupar el projecte. D'aquesta manera caldrà aconseguir una gestió del projecte suficientment bona per tal de complir els temps tot el que sigui possible.
- **Disseny d'una arquitectura adient:** Caldrà fer un estudi exhaustiu per tal de determinar l'arquitectura general que farem servir per al correcte desenvolupament de l'aplicació. També caldrà pensar en l'estructura de fitxers del projecte, per tal d'aconseguir un resultat final que sigui escalable, fàcil de millorar i entenedor.
- **Ús de diferents tècniques de testing i feedback d'usuaris per a millorar l'aplicació mòbil:** Una vegada haguem acabat el desenvolupament principal del projecte, es farà una primera Release al públic en general o a un cert nombre de persones (encara per definir). Amb el feedback dels usuaris recollit, s'arreglarien primer tots els bugs i desperfectes que presenta l'aplicació i seguidament s'implementarien totes les funcionalitats extres que els usuaris hagin trobat a faltar i que siguin viables.

## 5.2 Competències tècniques

Totes les competències tècniques utilitzades durant la realització d'aquest projecte han estat apreses al llarg de tot el Grau en Enginyeria Informàtica a la FIB, tant en la part inicial com en l'especialitat.

Ara es definiran les competències tècniques necessàries per el desenvolupament d'aquest projecte:

- **Obtenir els requisits d'un projecte:** Per tal d'aconseguir tota la informació necessària per a poder desenvolupar l'aplicació correctament, hauré d'obtenir els requisits de les persones implicades en el projecte, sobretot del creador i els usuaris finals.
- **Disseny d'arquitectura d'un software web:** El bon disseny de l'arquitectura de l'aplicació que es portarà a terme és un dels objectius més importants en aquest projecte. Per tant, necessitaré tots els coneixements adquirits en aquest àmbit per tal d'aconseguir una arquitectura que permeti assolir tots els objectius funcionals de l'aplicació.
- **Desenvolupament i manteniment d'un sistema de software amb connexió a internet:** Potser aquesta és la competència tècnica en què més s'aprofundirà en aquest projecte, ja que la part de desenvolupament de l'aplicació mòbil serà la més extensa.
- **Administració de Bases de Dades:** Caldrà utilitzar certs coneixements sobre administració de Bases de Dades, ja que en la part servidora guardarem totes les dades que generi la nostra aplicació, ja siguin els usuaris registrats, les partides jugades o que s'estàn jugant, diferents dades de resultats...
- **Gestió de Projectes de Software:** La gestió del projecte serà molt important per tal d'aconseguir un bon ritme de treball durant la realització del projecte i tenir cert control de les hores invertides en cada part del desenvolupament.
- **Testing i qualitat del producte:** Durant tot el període de desenvolupament es faran diferents tests de totes les features que estiguin en progrés. Quan acabi la fase de desenvolupament, es farà testing de tota l'aplicació final i s'intentarà millorar la qualitat d'aquesta.
- **Documentar correctament un projecte de Software:** Per a facilitar la possible millora futura del projecte, entre d'altres raons, serà molt important una documentació més o menys tècnica que permeti conèixer tots els punts importants d'aquest.

## 6 Planificació temporal

### 6.1 Planificació temporal

#### 6.1.1 Planificació estimada del projecte

Aquest projecte s'iniciarà el 1 de setembre de 2016 i tindrà com a data límit el 13 de gener, que és la data límit d'entrega de la memòria del TFG. En principi, el temps en que s'acabarà el projecte seràn unes 500 hores, repartides entre les diferents etapes d'aquest.

#### 6.1.2 Recursos

Els recursos (personals i materials) que s'utilitzaran durant la realització del projecte són:

- **Recursos personals:** una persona, que realitzarà tot el desenvolupament del projecte, és a dir, tant la part de ideació i disseny, com la de programació i construcció del software.
- **Recursos materials:**
  - Ordinador portàtil
  - Diversos dispositius mòvils Android.
  - Diversos dispositius mòvils iOS.

#### 6.1.3 Pla d'acció i valoració d'alternatives

A l'hora de realitzar el projecte es tindrà en compte un marge de unes 2 o 3 setmanes (20 dies) de retràs en la fase d'entrega final.

També s'ha de tenir en compte que es poden produir desviacions en les dates de les iteracions de desenvolupament. Així com en les tasques d'adquisició de coneixements.

La mesura per a solucionar aquestes desviacions, seria un possible redisseny de la planificació per a intentar ajustar de nou els temps estimats.

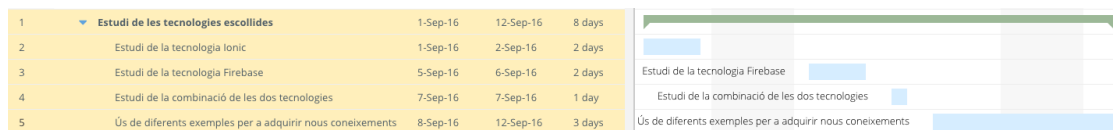
Tot i això, s'han estimat uns temps bastant amplis per tal de avançar-se a qualsevol possible desviació en la planificació inicial.

## 6.2 Descripció de les tasques

En aquest apartat es definiran les diferents tasques en què es divideix la planificació d'aquest projecte. Per a fer-ho, s'han utilitzat gràfics de Gantt.

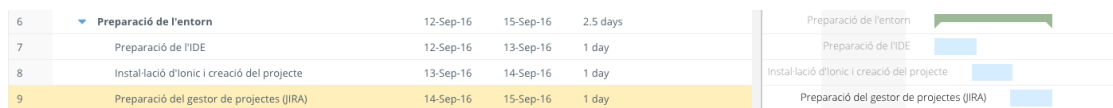
### 6.2.1 Estudi de la tecnologia escollida

En aquesta tasca estudiarem a fons les tecnologies que hem escollit finalment. En aquest cas, seràn el framework Ionic i Firebase. Per a fer-ho s'utilitzaran diferents tutorials i exemples per a formar-se en aquestes tecnologies.



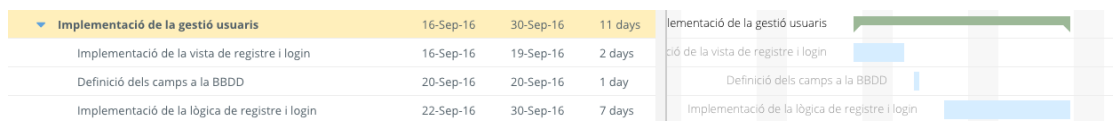
### 6.2.2 Preparació de l'entorn

Aquí, s'instal·larà tot l'entorn de desenvolupament i es prepararà el sistema de gestió de projectes escollit (JIRA).



### 6.2.3 Implementació de la gestió usuaris

En aquesta tasca implementarem tota la plataforma de gestió d'usuaris. És a dir, s'implementarà el sistema de login dels usuaris i el registre.



### 6.2.4 Implementació del joc

Aquesta serà la tasca més extensa i on es poden trobar més dificultats a l'hora d'acabar-la, ja que és sense cap mena de dubte la més complexa. Consistirà en implementar el joc en si. Això comportarà la implementació del taulell de joc, les tirades dels usuaris, la interacció entre ells...

14	▼ Implementació del joc	3-Oct-16	28-Nov-16	41 days	
15	Disseny del tauler de joc	3-Oct-16	14-Oct-16	10 days	
16	Interacció entre els usuaris durant la partida	17-Oct-16	10-Nov-16	19 days	
17	Altra lògica de la partida	14-Nov-16	28-Nov-16	11 days	

### 6.2.5 Retocs finals

En aquesta etapa acabarem de definir l'estil final de l'aplicació, com també implementarem les últimes features que entraran a la primera release d'aquesta.

18	▼ Retocs finals	29-Nov-16	30-Dec-16	24 days	Retocs finals
19	Estil de l'aplicació	29-Nov-16	9-Dec-16	9 days	de l'aplicació
20	Últims retocs a la lògica de l'aplicació	12-Dec-16	30-Dec-16	15 days	Últims retocs a la lògica de l'aplicació

### 6.2.6 Primera Release de l'aplicació i entrega de la documentació

Finalment, a la última etapa, recopilarem tota la documentació que s'hagi anat generant durant tot el desenvolupament. Després es farà la primera release de l'aplicació als mercats mòbils més coneguts (Apple Store i Google Play).

Primera Release	2-Jan-17	13-Jan-17	10 days		Primera Release
-----------------	----------	-----------	---------	--	-----------------

### 6.2.7 Desviacions en la planificació

Durant la fase de desenvolupament del projecte s'ha produït alguna desviació important, que ha produït algun canvi en la planificació temporal.

Això ha succeït sobretot en la part d'implementació del joc, on m'he trobat amb molts problemes i obstacles que van fer que s'haguessin d'invertir moltes hores en solucionar-los.

Per aquest motiu, algunes features no rellevants no s'han pogut implementar dins d'aquest projecte. A més a més, es va ampliar el temps destinat al desenvolupament, reduint considerablement el temps de testing del software i el de resolució de bugs i consolidació.

## 7 Gestió econòmica

### 7.1 Identificació dels costos

Els costos del projecte estan relacionats amb els recursos que es descriuen a continuació.

Per a fer la gestió dels gastos classifiquem els recursos en recursos personals i recursos materials.

- **Recursos personals:** una persona, que realitzarà tot el desenvolupament del projecte, és a dir, tant la part de ideació i disseny, com la de programació i construcció del software.
- **Recursos materials:**
  - Ordinador portàtil
  - Diversos dispositius mòbils Android.
  - Diversos dispositius mòbils iOS.

### 7.2 Estimació dels costos

#### 7.2.1 Costos directes per tasques

Encara que aquest projecte no es porti a terme en una empresa, posarem un sou per hora realista per estimar els costos directes d'aquest.

El projecte serà desenvolupat per una sola persona, per tant, aquesta mateixa persona serà tant el project manager, com l'analista programado, desenvolupador i tester.

La remuneració que associem a cada rol correspon amb el preu per hora mínim que estableix l'informe de Page Personnel de 2016. Utilitzant aquestes dades, s'han calculat els següents sous:

- Project Manager 18€/h
- Analista programador: 12€/h
- Desenvolupador: 9€/h
- Tester: 11 €/h

Amb aquests sous s'han calculat els següents costos directes:



	Dedicació				
Fase	Project Manager	Analista Programador	Desenvolupador	Tester	Hores totals
Estudi de la tecnologia escollida	10h	40h	50h		100h
Preparació de l'entorn	10h	5h	5h		20h
Implementació de la gestió d'usuaris		10h	30h	10h	50h
Implementació del joc		50h	150h	50h	250h
Retocs Finals		20h	40h	40h	100h
Primera Release	10h	5h	10h	5h	30h
Preu per hora	18 €	12 €	9 €	11 €	
Preu total	540 €	1.560 €	2.565 €	1.155 €	

El cost directe total serà de 5820 €.

### 7.2.2 Costos indirectes

Dins dels costos indirectes tindrem en compte el hardware que utilitzarem (la seva amortització i el cost d'aquest) i el software utilitzat durant el desenvolupament del projecte.

Els diferents costos indirectes són aquests:

- **Amortització del hardware:** Durant el desenvolupament d'aquest projecte s'utilitzaran diversos dispositius per a validar que l'aplicació funciona perfectament.
  - **Ipad Air 1:** Ens servirà per a testejar l'aplicació en el sistema operatiu mòvil iOS. Aquest dispositiu té un cost de 399 €.
  - **One Plus X:** Ens servirà per a testejar l'aplicació en el sistema operatiu mòvil Android. Aquest dispositiu té un cost de 249 €.
  - **MacBook Air:** Ens servirà per a dissenyar, desenvolupar i testejar tota l'aplicació, a part de gestionar el projecte i documentar-lo. El seu cost és de 1200 €.

Com que cap d'aquests dispositius s'ha comprat específicament per a aquest projecte, només comptabilitzarem un 10% del cost total d'aquests cada mes.

- **Amortització del software:** En principi, s'utilitzaran llicències d'estudiant per a tot el que sigui necessari (Jira, IDE...), per tant, no haurem de comptabilitzar cap cost del software utilitzat.

A continuació es mostra la taula dels costos indirectes que s'han produït durant el projecte:

Cost	Mesos	Preu unitari	Cost Total
Amortització Ipad Air 1	5	39.9 €	199.5 €
Amortització One Plus X	5	24.9 €	124.5 €
Amortització MacBook Air	5	120 €	600 €
			924 €

### 7.2.3 Pressupost final

Després de fer l'estudi dels diferents tipus de costos, hem obtingut la següent taula:

Concepte	Cost
Costos Directes	5.820 €
Costos Indirectes	924 €
	6.744 €

## 7.3 Control de gestió

Ara per ara, el control de gestió de costos d'aquest projecte no serà completament correcte. Ja que les hores que s'invertiran en aquest poden variar considerablement.

Per això, s'enregistraran totes les hores de desenvolupament, per tal de aconseguir un cost del projecte correcte. Així veurem si la nostra estimació s'acostava al que inicialment s'ha estimat.

## 8 Sostenibilitat

En aquest apartat es valorarà la Sostenibilitat (Econòmica, Social i Ambiental) del projecte. Per a fer-ho, es valoraran diversos apartats que ja ens venen definits. Aquesta és la valoració que tindria aquest projecte:

Sostenibilitat	Econòmica	Social	Ambiental
Valoració	6,6	7	7

### 8.1 Econòmica

En el cas de la sostenibilitat econòmica, cal destacar l'avaluació dels costos que s'ha fet prèviament al desenvolupament del projecte. A part d'això, les hores de desenvolupament han estat estimades de la millor manera possible i tant ajustades com ha sigut possible. El temps que s'ha dedicat a cada tasca és proporcional a la seva importància dins del projecte.

### 8.2 Social

Pel que fa a la sostenibilitat social, aquest projecte està enfocat a usuaris de qualsevol edat i, encara que no tingui una utilitat real, aquests es podran distreure utilitzant l'aplicació.

### 8.3 Ambiental

La sostenibilitat ambiental no podria ser millor en aquest projecte, ja que no s'utilitzarà cap mètode de transport per el desenvolupament del projecte, i tampoc es necessitarà un consum excessiu d'electricitat per portar-lo a terme.

## 9 Especificació

### 9.1 Requisits

#### 9.1.1 Requisits d'interfície

Aquests requisits descriuràn, de forma més o menys detallada, totes les entrades i sortides del sistema de software.

##### Interfície d'usuari

Des d'un bon principi es van estudiar els colors i les formes del taulell per tal de que l'adaptació a aplicació mòbil fos del gust del creador del joc. Els únics requisits donats, van ser només sobre el taulell de joc:

- Les fitxes que utilitzarem en el joc hauràn de ser atractives i dels colors donats. No existiràn fitxes diferents a les del joc de taula real, és a dir, amb valors diferents del 1 al 15 i el joker.
- El taulell de l'aplicació, emularà un taulell quadrat de fusta.

En quant a l'aplicació en si, haurà de ser fàcil i intuïtiva d'utilitzar i no serà necessari una gran ajuda per a què l'usuari aprengui a jugar i fer-ne ús. Per tant, caldrà cuidar molt tots els detalls d'usabilitat i millorar al màxim l'experiència d'usuari.

##### Interfície de hardware

Els requisits de hardware que han de complir els usuaris de la nostra aplicació són:

- El dispositiu ha de suportar iOS o Android.
- El dispositiu necessita de connexió a internet per a la total utilització de l'aplicació.
- Caldrà que el sistema tingui suficient espai d'emmagatzematge per tal de poder instal·lar l'aplicació.

##### Interfície de comunicació

En aquest projecte hi hauràn diversos requisits que faràn referència a la comunicació entre els diferents jugadors del sistema:

- El lobby on apareixen les partides actives del jugador s'haurà de refrescar automàticament i en *real-time*, per tal de millorar l'experiència de l'usuari.

- El xat entre els jugadors d'una mateixa partida també es refrescarà automàticament i en *real-time*.

### 9.1.2 Requisits funcionals

Els requisits funcionals definiran accions fonamentals que el nostre sistema ha de realitzar al rebre informació, processar-la i, posteriorment, produir certs resultats. Aquí llistarem les comprovacions generals més importants que es produeixen durant l'execució de l'aplicació:

- Comprovació de registre i autenticació correctes.
- Comprovació a l'hora de crear una partida. Els paràmetres d'aquesta partida han de ser correctes.
- Comprovar que el taulell és correcte abans de processar les dades.

### 9.1.3 Requisits no funcionals

#### Rendiment

El rendiment de l'aplicació serà fonamental per tal de que l'experiència de l'usuari sigui bona.

El punt més crític, on el rendiment serà d'allò més important, és a l'hora de fer una jugada al taulell. Tots els moviments de fitxes del taulell han de ser el més ràpids possibles.

#### Seguretat

La seguretat en l'aplicació serà molt important en la part del registre i autenticació dels usuaris.

Cal assegurar la correcta encriptació de les dades privades de l'usuari, com la contrasenya, per evitar qualsevol risc.

## 9.2 Casos d'ús de l'aplicació mòbil

En aquest apartat llistarem tots els casos d'ús que ha de complir la versió final de l'aplicació mòbil.

### 9.2.1 Casos d'ús relacionats amb l'autenticació d'usuari

- **Cas d'ús 1:** Un usuari no existent es vol registrar al sistema.
- **Cas d'ús 2:** Un usuari existent vol loguejar-se al sistema.

### 9.2.2 Casos d'ús Lobby

- **Cas d'ús 3:** Veure les partides actives en el moment actual.
- **Cas d'ús 4:** Veure les partides en què li toca tirar a l'usuari.
- **Cas d'ús 5:** Anar al formulari de creació de partida.
- **Cas d'ús 6:** Jugar en una partida en concret.

### 9.2.3 Casos d'ús Settings

- **Cas d'ús 7:** Veure el tutorial de joc.
- **Cas d'ús 8:** Veure informació adicional sobre l'aplicació (nom del creador, versió...).

### 9.2.4 Casos d'ús Ranking

- **Cas d'ús 9:** Veure Ranking global del joc.

### 9.2.5 Casos d'ús User

- **Cas d'ús 10:** Veure imatge de perfil de l'usuari connectat.
- **Cas d'ús 11:** Veure la informació de l'usuari connectat.
- **Cas d'ús 12:** Veure un gràfic representatiu de les partides guanyades i perdudes de l'usuari connectat.

### 9.2.6 Casos d'ús Creació de partida

- **Cas d'ús 13:** Seleccionar un nom per la partida que es crearà.
- **Cas d'ús 14:** Seleccionar el número de jugadors que participaran a la partida.
- **Cas d'ús 15:** Seleccionar els jugadors que participaran a la partida.
- **Cas d'ús 16:** Seleccionar el número de pals que s'utilitzaran a la partida.
- **Cas d'ús 17:** Seleccionar el número de jokers que hi hauran a la partida.
- **Cas d'ús 18:** Crear partida amb els paràmetres seleccionats.

### 9.2.7 Casos d'ús Joc

- **Cas d'ús 19:** Veure taulell de joc.
- **Cas d'ús 20:** Veure les fitxes que té l'usuari connectat.
- **Cas d'ús 21:** Maximitzar i minimitzar el taulell per veure'l amb més o menys detall.
- **Cas d'ús 22:** Tirar una fitxa de l'usuari al taulell .
- **Cas d'ús 23:** Retornar una fitxa que es trobi en el taulell i que el mateix usuari hagi tirat en el mateix torn.
- **Cas d'ús 24:** Moure qualsevol fitxa que es trobi en el taulell de joc.
- **Cas d'ús 25:** Tornar a l'estat inicial del taulell, retornant les fitxes de l'usuari i col·locant les fitxes del taulell tal i com les ha deixat el jugador del torn anterior.
- **Cas d'ús 26:** Passar el torn sense fer cap canvi en el taulell.
- **Cas d'ús 27:** Jugar amb els canvis que s'hagin fet en el taulell.
- **Cas d'ús 28:** Comprovar que el taulell estigui correcte abans de passar el torn al següent jugador.
- **Cas d'ús 29:** Avisar al jugador si la seva jugada no és correcta i, per tant, ha de canviar-la per a poder jugar.
- **Cas d'ús 30:** Accedir al xat amb els jugadors de la partida.

### 9.2.8 Casos d'ús Xat

- **Cas d'ús 31:** Veure tots els missatges enviats pels jugadors de la partida.
- **Cas d'ús 32:** Enviar un missatge a la resta de jugadors de la partida.

## 9.3 Casos d'ús de l'Administrador Web

Aquí es definiran els diferents casos d'ús de l'administrador web.

Com es pot veure, aquesta aplicació web, a part de les analítiques, serà un **CRUD** (*Create, Read, Update and Delete*) del nostre sistema.

- **Cas d'ús 1:** Veure tots els usuaris creats en el sistema.
- **Cas d'ús 2:** Crear un usari.
- **Cas d'ús 3:** Esborrar un usuari del sistema.
- **Cas d'ús 4:** Veure totes les partides creades en el sistema.
- **Cas d'ús 5:** Crear una partida.
- **Cas d'ús 6:** Esborrar una partida del sistema.
- **Cas d'ús 7:** Veure usuaris actius en el sistema a temps real.
- **Cas d'ús 8:** Veure partides actives (no acabades) en el sistema.
- **Cas d'ús 9:** Veure diferents estadístiques de usuaris i partides en el sistema.



## 9.4 Model Conceptual del Sistema

Abans de començar la fase de desenvolupament es va plantejar com hauria d'organitzar-se el sistema per tal de tenir una estructura de dades prou modular i que ens permeti una certa escalabilitat en el futur.

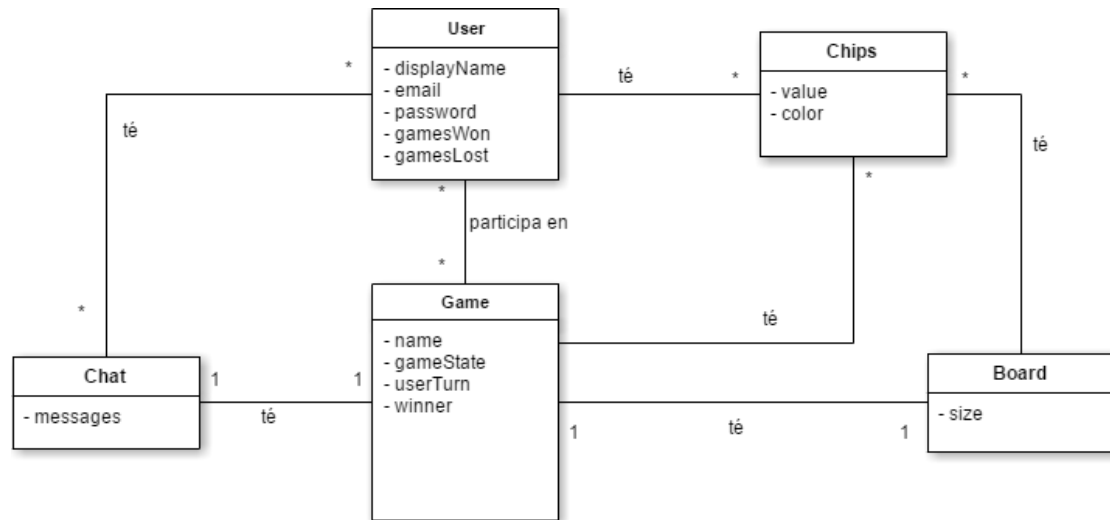


Figura 17: Model Conceptual del sistema

Els usuaris del sistema, podran participar en tantes partides com vulguin en qualsevol moment. En aquestes partides hi participaran com a mínim 2 usuaris i cadascuna tindrà associat un taulell de joc i un xat entre els participants.

Les fitxes (*Chips*) de la partida estaran repartides entre els participants, el taulell i la mateixa partida. Aquestes fitxes tindran un valor i un color determinats, a més a més d'un identificador únic, ja que depenent de les fitxes amb les que juguem n'hi poden haver de repetides.

Els usuaris de l'aplicació tindran tants xats com partides actives a les quals participin. Una vegada la partida s'hagi acabat, és a dir, que algun jugador hagi guanyat, el xat corresponent de la partida serà eliminat i els jugadors ja no podran enviar missatges a aquest en concret.

Tot i que aquest model conceptual dóna una idea de com serà el resultat final del sistema, és només un primer esbós d'aquest. Per tal d'entendre bé com funciona tot, caldrà estudiar bé en l'estructura de la Base de Dades no relacional, el mapa navegacional i la interfície gràfica del sistema final.

## 10 Disseny del Sistema

### 10.1 Arquitectura Global del Sistema

En aquest apartat tractarem l'arquitectura del nostre sistema.

#### 10.1.1 Patró de disseny : MVC

El patró de disseny que s'ha escollit per a aquest projecte ha estat el patró de **Model Vista Controlador** o **MVC**, ja que és el patró més extès per a desenvolupar aplicacions web i ens permet obtenir un resultat molt modular que ens servirà per a millorar l'escalabilitat de l'aplicació entre d'altres propietats.

El patró MVC està format per tres parts:

- **Model** : És la capa inferior del patró responsable de la definició de les dades.
- **Vista** : És la capa responsable de mostrar tota les dades a l'usuari.
- **Controlador** : És la part de codi que conté totes les interaccions entre el Model i la Vista.

A part d'això, utilitzarem **Services**, que ens serviràn per obtenir, modificar i inserir dades a la Base de Dades del nostre sistema i que l'accés a aquesta no es faci desde el controlador directament. Per tant, els Services ens ompliràn els models definits al controlador, per a què aquest els utilitzi, internament o mostrant-los a l'usuari en qualsevol vista.

En aquest projecte, s'ha decidit que cada vista definida tingui un controlador específic. Això ens ha permès millorar molt la modularitat i entenibilitat del projecte, fent així que sigui molt fàcil afegir noves *features* a l'aplicació.

Les vistes de l'aplicació estaràn definides en diferents *templates* en HTML. Cada vista, tindrà la seva *template* específica.

Seguidament, es mostra un petit esquema de l'arquitectura global que s'ha seguit en el desenvolupament d'aquesta aplicació.

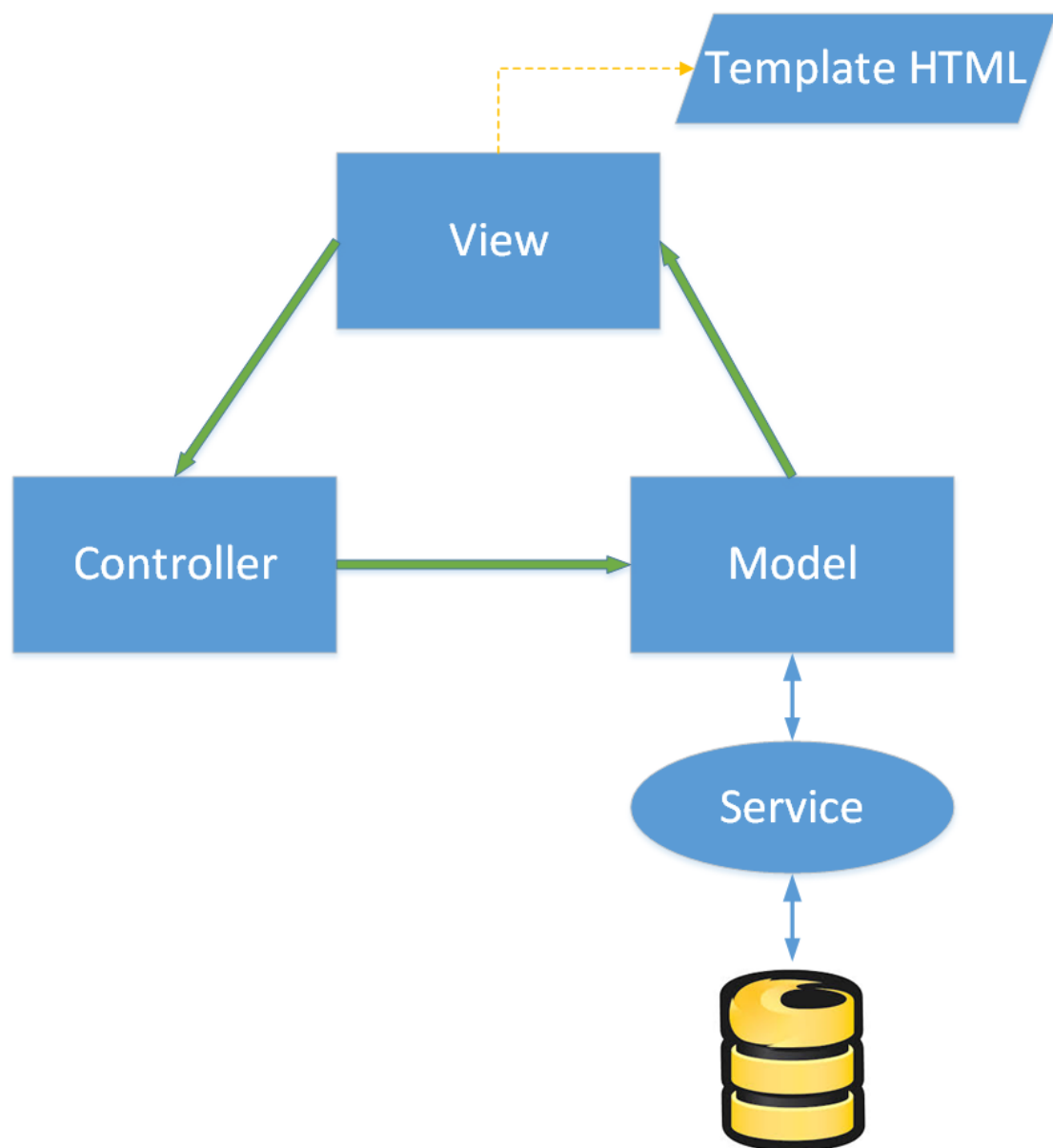


Figura 18: Esquema de l'arquitectura MVC

## 10.2 Mapa navegacional

Per a entendre correctament com funcionarà l'aplicació, serà important dissenyar un bon mapa navegacional abans de començar el desenvolupament i retocar-lo quan el projecte vagi avançant.

En la figura de la pàgina següent es pot veure aquest mapa navegacional.

La primera vista que se li mostrarà a l'usuari al iniciar el software serà la pàgina de login, on els usuaris ja registrats podran loguejar-se i els nous hauràn de passar pel formulari de registre.

Una vegada loguejats els usuaris seràn redirigits a la *InitialPage*, que és la pàgina general que conté totes les pàgines principals ordenades en 4 pestanyes. Inicialment la *InitialPage* ens mostrarà la pàgina de *Lobby*, on es llisten totes les partides actives dels l'usuaris.

A partir d'aquest punt, l'usuari ja podrà navegar lliurement per l'aplicació.

Per tant podrà anar a qualsevol de les pestanyes de la pantalla principal (*Settings*, *Lobby*, *Ranking* i *UserInfo*), i també podrà crear una nova partida i jugar a qualsevol partida activa que tingui.

Quan l'usuari vagi a la pàgina de creació de partida, haurà d'omplir el formulari especificant tots els paràmetres necessaris per a parametritzar correctament la partida.

Per a seleccionar els jugadors de la partida, l'usuari podrà accedir a la pàgina de *AddPlayers*, on es llistaràn els *displayName's* de tots.

Per a representar totes aquestes relacions entre les diferents vistes de l'aplicació, s'ha dissenyat el següent mapa navegacional:



## 10.3 Tecnologia i eines usades pel desenvolupament

### 10.3.1 Ionic Framework

Ionic és un SDK de HTML5 molt potent que ens permet compilar aplicacions per a dispositius mòbils utilitzant tecnologies web com HTML, CSS i Javascript.

Encara que el resultat no és el d'una aplicació nativa, Ionic intenta simular els comportaments nadius de cada sistema operatiu mòbil (iOS o Android).

El que fa Ionic a l'hora de compilar per a cada sistema operatiu és crear un projecte en la plataforma en concret i, seguidament, inserir un Web View que mostrarà la nostra aplicació desenvolupada amb tecnologies web. És per això que Ionic només és compatible amb versions de sistemes operatius que suportin els Web Views en aplicacions natives.

A iOS, el dispositiu haurà de tenir una versió o igual a iOS 7. En el cas d'Android, necessitem que el dispositiu tingui com a mínim la versió 4.1.

Actualment, Ionic requereix AngularJS per a que poguem exprimir tot el seu potencial. Això ens facilitarà l'ús de l'arquitectura web que més ens convingui.

A més a més, Ionic ens abastirà de components CSS, com headers, botons, llistes, formularis... que emulen l'estil de les aplicacions natives de cada sistema operatiu mòbil.

Per a facilitar la feina del desenvolupador a l'hora de crear, compilar i emular un projecte de Ionic, es fa servir la CLI (Command Line Interface) específica del framework. Aquesta s'utilitzarà desde la terminal del sistema operatiu on estem treballant.

Una de les característiques més útils d'aquesta CLI, serà la de emular des del navegador el comportament que tindrà la nostra aplicació en el dispositiu mòbil.

Tot i que això no és una prova fiable de que l'aplicació funcionarà perfectament en el dispositiu mòbil, ens ajudarà molt a l'hora de desenvolupar sense la necessitat de compilar cada vegada que volgum veure el resultat del nostre codi.

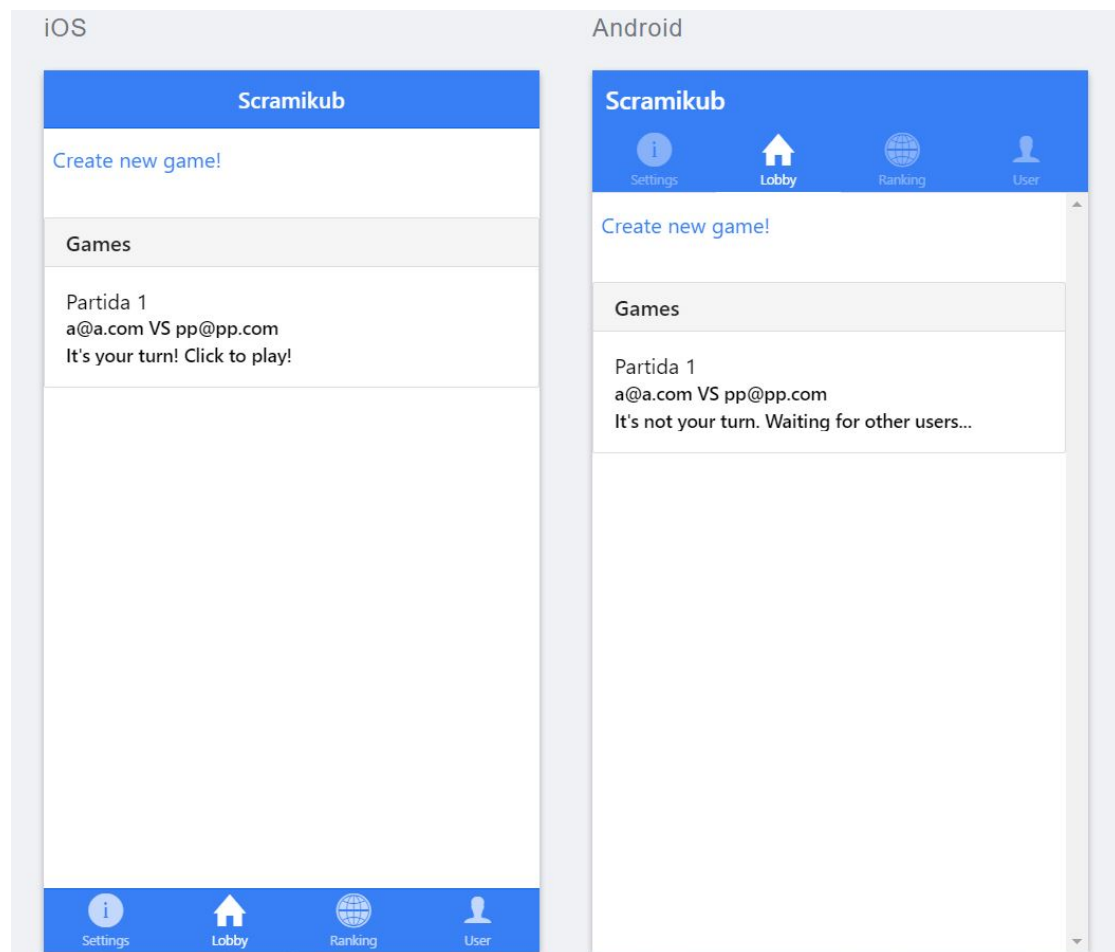


Figura 20: Emulador CLI Ionic

En el cas de la captura anterior, s'ha utilitzat la comanda *ionic serve -lab*, i ens permet emular l'aplicació per a Android i iOS alhora.

Ionic és un SDK *open source* llançat sota la llicència *MIT*. Això significa que podem utilitzar aquesta eina tant per a l'ús personal com per a projectes comercials de manera gratuïta.

La llicència *MIT* és utilitzada a molts projectes importants com *jQuery* i *Ruby on Rails*.

### 10.3.2 Firebase

Firebase és un servei que ens permet emmagatzemar i obtenir dades d'una manera molt senzilla com si fos una API convencional, sense necessitat de desenvolupar-la. Les dades s'emmagatzemen dins de contenidors en format JSON gestionats per un sistema no relacional o NoSQL, a l'estil de MongoDB. Aquestes dades poden ser gestionades mitjançant la consola que ens ofereixen a través de la seva web.



Figura 21: Logo Firebase

Les dos característiques més interessants d'aquest servei són la sincronització de dades en temps real i la sincronització de dades sense connexió. La sincronització de dades en temps real ens permetrà que si, per exemple, un usuari modifiqués una dada determinada, aquesta mateixa dada serà actualitzada per a la resta de usuaris en temps real.

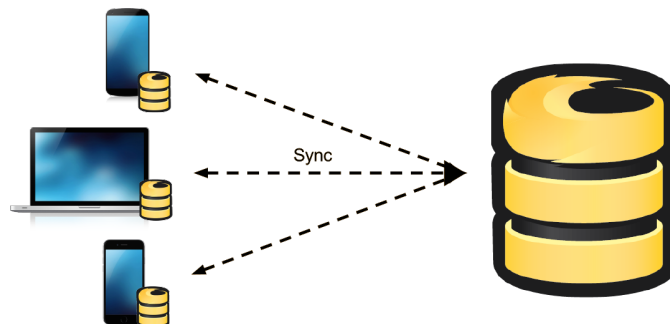


Figura 22: Esquema de sincronització entre dispositius i Firebase

L'altra funcionalitat permetrà als usuaris, modificar qualsevol dada encara que per qualsevol motiu es trobin sense connexió. Firebase s'encarregarà d'emmagatzemar les dades en memòria local i ho sincronitzarà amb la base de dades quan l'aplicació torni a tenir connexió.



## 10.4 Interfície gràfica

En aquest apartat es mostrarà i es defensarà la interfície gràfica de les pantalles més importants de l'aplicació.

Com s'ha dit anteriorment, s'ha apostat per una interfície simple per tal de facilitar l'ús d'aquesta. Tot i això, el nombre de funcionalitats disponibles és molt gran.

### 10.4.1 Login i Registre

Les pàgines de login i registre de l'aplicació s'han dissenyat per a que siguin totalment intuïtives i molt fàcils d'utilitzar.

The image displays two side-by-side screenshots of an application's user interface for login and registration.

**Figure 23: Pàgina de login (Android)**

The login screen features the "SCRAB" logo at the top, which includes a 3D cube icon. Below the logo are two input fields: "Username" and "Password". A prominent blue "Login" button is positioned below these fields. At the bottom, there is a link labeled "Create New Account" accompanied by a small user icon.

**Figure 24: Pàgina de registre (Android)**

The registration screen has a blue header bar containing the text "Create New Account" and a "Cancel" button. Below the header are three input fields: "Email Address", "Passsword", and "Display Name". A blue "Create" button is located at the bottom of the form.

Figura 23: Pàgina de login (Android)    Figura 24: Pàgina de registre (Android)

En el cas de que l'usuari estigui registrat, només haurà d'introduir el seu email i

contrasenya en els camps corresponents. En el cas de que les dades introduïdes siguin correctes, l'usuari serà redirigit cap al *Lobby*. En el cas contrari, apareixerà una alerta informant de que algun dels camps no s'ha omplert correctament.

Si l'usuari necessita registrar-se, només haurà de prémer el botó per a crear un nou compte i omplir el formulari de la Figura 24. En el cas de que l'usuari no ompli correctament algun camp o el email introduït ja existeixi en el sistema, es mostrarà una alerta.

#### 10.4.2 Lobby

Aquesta pantalla serà la primera a la qual l'usuari serà redirigit just després d'autenticar-se correctament

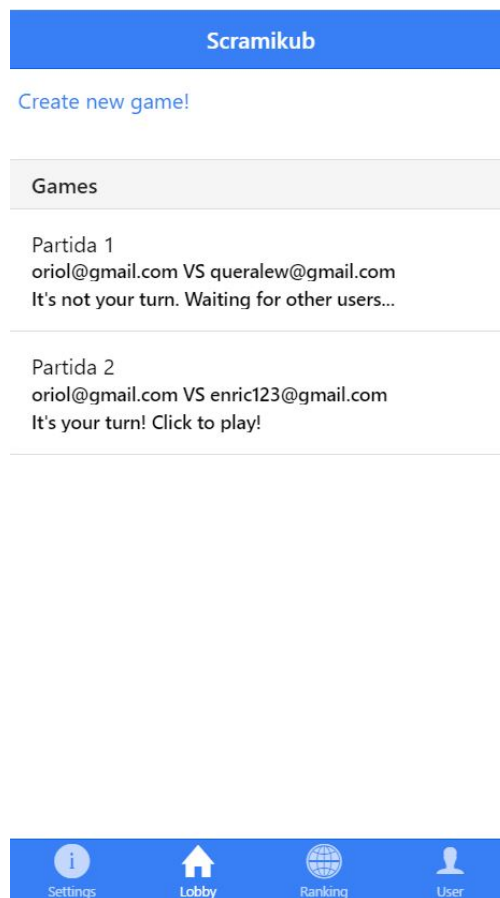


Figura 25: Lobby de l'aplicació (iOS)

Aquí només s'hi llistaran les partides actives de l'usuari connectat. De cada partida veurem el nom d'aquesta, els jugadors que hi participen i un missatge

indicant si és el torn de l'usuari connectat o no.

Una vegada un usuari esculli una partida per a jugar, si és el seu torn, accedirà a la pantalla del taulell de joc on podrà fer la seva jugada. Si no és el seu torn, apareixerà un missatge indicant que cal que s'esperï per a jugar la partida.

A part d'això, a la part superior de la pantalla s'ens dóna la possibilitat de crear la nostra pròpia partida.

### 10.4.3 Informació d'usuari

En aquesta pantalla es mostrarà la informació de l'usuari connectat d'una manera molt senzilla.

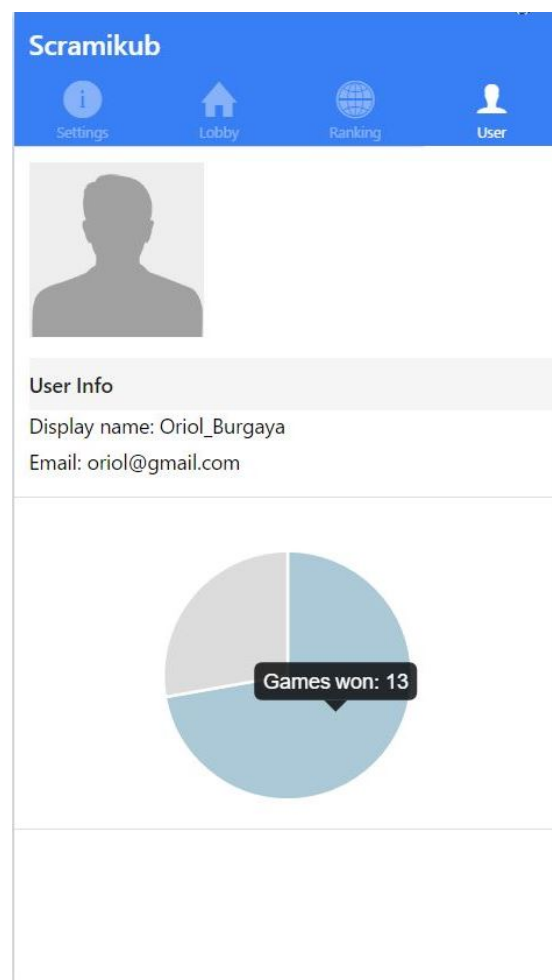


Figura 26: Pàgina d'informació de l'usuari (Android)

Com es pot observar en la figura anterior, es mostrarà tant el *Display name*

de l'usuari com el seu *email*. Finalment es mostrarà un gràfic de les partides guanyades i perdudes d'aquest usuari. Quan es pulsi sobre la zona de partides guanyades en el gràfic, apareixerà una etiqueta com es mostra a la figura anterior.

#### 10.4.4 Creació de partida

La següent pantalla ens permetrà parametritzar una nova partida abans de, finalment, crear-la.

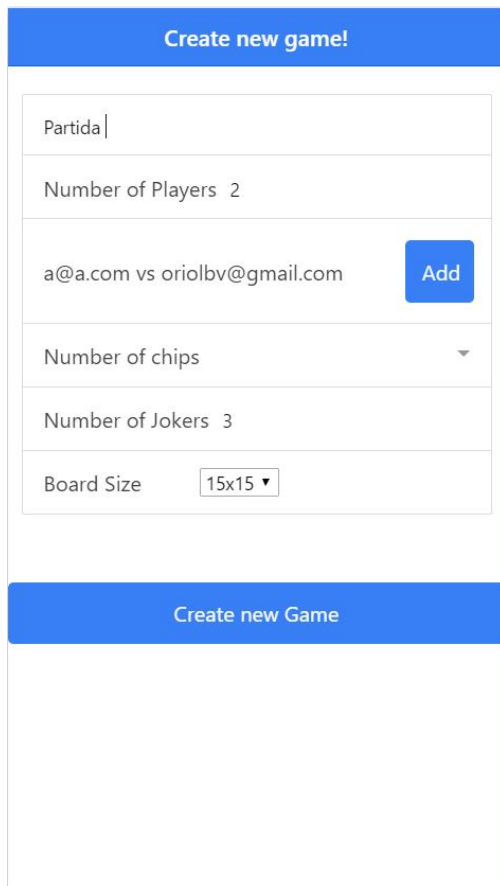


Figura 27: Pàgina de creació de partida (iOS)

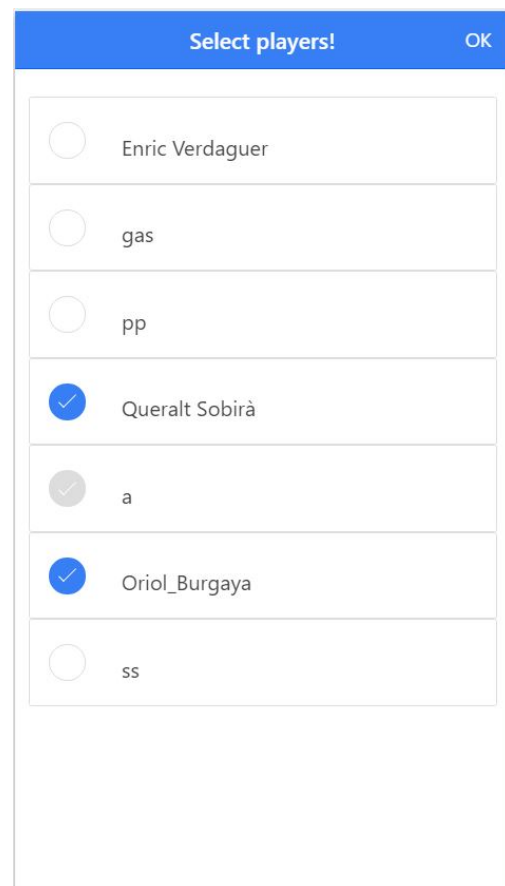


Figura 28: Pàgina de selecció de jugadors (iOS)

En aquesta pantalla, l'usuari introduirà tots els paràmetres necessaris per a personalitzar i crear una nova partida:

- **Nom de la partida:** Nom per a identificar la partida. No pot ser el mateix que el nom d'una altra partida activa actualment.
- **Número de jugadors:** Número de jugadors que participaran a la partida. Aquest número ha d'estar entre 2 i 8. El màxim número de jugadors ve donat pel nombre màxim de fitxes que es poden repartir durant una partida.
- **Jugadors:** Per a seleccionar els jugadors que participaran a la partida, haurem de prémer el botó *Add*, que ens redirigirà a la pantalla de selecció d'usuaris.  
Una vegada entrem a aquesta pantalla, l'usuari connectat ja estarà seleccionat, ja que aquest haurà de participar a la partida que crearà. Seguidament, s'hauran de seleccionar tants jugadors com els que hàgim indicat a la pàgina de creació.  
Si s'han seleccionat correctament els jugadors de la partida, al prémer *Ok* en la part superior de la pantalla, l'usuari serà retornat a la pàgina de creació de la partida per a seguir omplint els camps de configuració.
- **Número de fitxes:** En aquest camp es decidirà el nombre de fitxes que es repartiran durant tota la partida.  
Els números de fitxes estan predefinits i l'usuari pot triar entre aquests: 60, 90, 120, 180.
- **Número de jokers:** Aquí es definirà el número de jokers que hi hauran a la partida.  
Aquest número no està limitat de cap manera, però si és excessiu pot influir molt en l'experiència de joc.
- **Mida del taulell:** En aquest camp l'usuari podrà indicar la mida que farà el taulell de joc durant la partida. Aquesta opció influirà molt en la jugabilitat de la partida.  
Cal dir que en la versió final de l'aplicació aquesta funcionalitat estarà en fase beta, ja que no s'ha pogut testejar degudament.

Una vegada tots els camps estiguin omplerts, l'usuari podrà crear la partida utilitzant el botó de la part inferior del formulari.

### 10.4.5 Taulell de joc

Ara s'explicarà tota la part relacionada amb l'interfície gràfica de la pantalla on hi ha el taulell de joc de la partida.



Figura 29: Taulell sense zoom (Android)

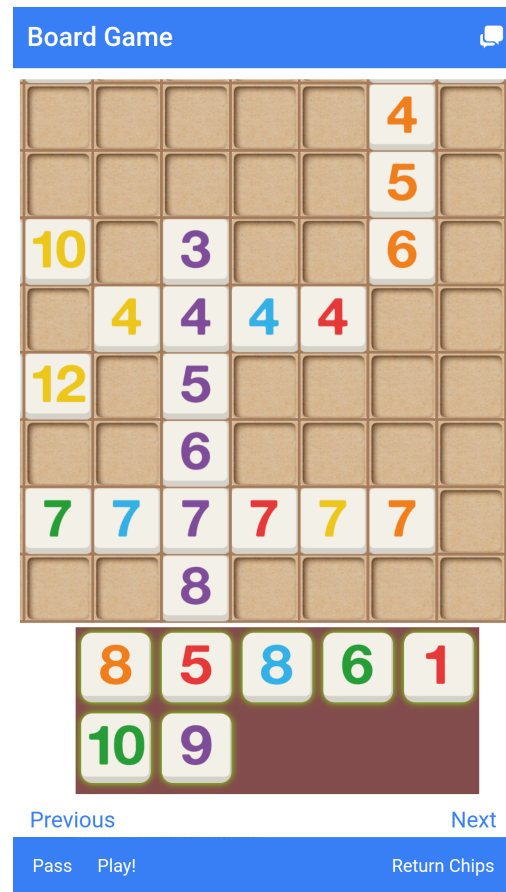


Figura 30: Taulell amb zoom (Android)

A la part central de la pantalla es pot veure el taulell (en aquest cas, de 15x15), en el qual podrem fer zoom i scroll (vertical i horitzontal) per moure'ns tranquil·lament i veure-ho tot amb la precisió que vulguem.

Les fitxes que ja es trobin en el taulell, es podran canviar de lloc per a què el jugador faci la seva jugada fent click a la fitxa que es vol canviar de posició i un altre click a la posició destí.

Just a sota del taulell, hi trobem totes les fitxes del jugador en dos files de com a mínim 5 columnes. En el cas de que no es vegin totes les fitxes a la pantalla,

l'usuari pot utilitzar els botons *Next* i *Previous*, per a veure la resta de fitxes que té.

Per a jugar una fitxa, l'usuari només ha d'arrossegar-la cap a la posició del taulell que vulgui (*drag*) i deixar-la anar (*drop*). Cal tenir en compte, que si no s'ha fet zoom al taulell, pot ser que quan deixem anar la fitxa sobre el taulell no es col·loqui a la posició desitjada, per tant, caldrà recol·locar-la.

A la part inferior de la pantalla ens trobarem tres botons bastant importants:

- **Pass:** Ens servirà per a passar el torn en el cas de que no poguem fer cap jugada. El taulell tornarà al seu estat originial i se li repartirà una fitxa aleatòria al usuari en concret. Després l'usuari serà redirigit al Lobby.
- **Play:** L'aplicació farà totes les comprovacions per saber si el taulell és correcte. En el cas de que ho sigui, la jugada serà tramesa i l'usuari serà redirigit al Lobby. En el cas de que la jugada no sigui correcta, es mostrarà una alerta i l'usuari no serà redirigit enlloc.
- **Return Chips:** Retorna el taulell a l'estat inicial i es tornen també totes les fitxes de l'usuari.

Finalment, a la part superior de la pantalla hi haurà el botó per accedir al xat de la partida, on podrem xatejar amb la resta de jugadors d'aquesta.

#### 10.4.6 Xat

La pantalla de xat serà utilitzada per a comunicar-se entre tots els jugadors d'una partida concreta.

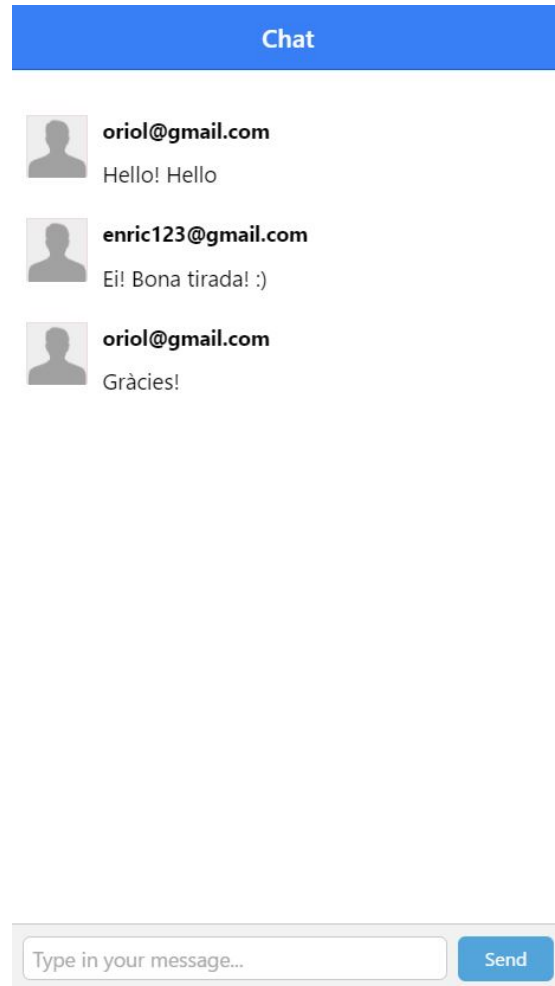


Figura 31: Pàgina de xat d'una partida (iOS)

En aquesta pantalla es llistaràn tots els missatges enviats entre els jugadors de la partida amb l'estructura de la figura anterior.

A la part inferior de la pantalla, l'usuari disposarà d'un camp per a escriure el missatge i un botó per a enviar-lo a la resta de jugadors. Una vegada enviat, el missatge es veurà automàticament al llistat de missatges.



## 10.5 Integració en varis Sistemes Operatius

Un dels dubtes que es van generar al iniciar el projecte, va ser si seria possible integrar la nostra aplicació en els dos sistemes operatius mòbils amb els que treballlem.

Cal tenir en compte que, tot i l'ajuda que ens donen els components de Ionic, desenvolupar una aplicació per ambdós sistemes operatius no és tant fàcil com fer una aplicació nativa per a un d'ells.

Durant el desenvolupament del projecte s'han hagut de fer molts *workarounds* per tal de que tots els components gràfics i gestos nadius funcionessin correctament en ambdós sistemes operatius mòbils. També s'ha hagut de vigilar molt amb temes de rendiment de l'aplicació, ja que, sobretot en alguns dispositius Android, es notava una certa caiguda d'aquest en la pantalla del taulell de joc.

Tots aquests problemes s'han anat solucionant a partir d'hores i hores, d'investigació i desenvolupament.

A part, el bon disseny inicial del projecte, ha ajudat molt en aconseguir que la integració en els dos sistemes operatius mòbils fos prou satisfactòria.

En el cas de la interfície gràfica, he intentat utilitzar un estil que es troba entre una aplicació d'iOS i una d'Android. Tot i això, el resultat final de l'aplicació té més semblances amb una aplicació nativa d'iOS que no pas amb una d'Android, en quant a interfície gràfica.

## 10.6 Usabilitat

Ara, s'intentaràn resumir tots els aspectes que s'han tingut en compte per a millorar al màxim l'usabilitat de l'aplicació.

Per a fer un petit estudi d'usabilitat de l'aplicació i millorar-lo en el cas de que el resultat d'aquest estudi fos negatiu, s'han utilitzat varis atributs importants a l'hora de determinar si un software per a dispositius mòbils és usable o no.

Evidentment, el resultat final d'aquest estudi serà positiu, ja que l'utilitzaré per millorar tots els aspectes presentats.

Usabilitat de l'aplicació	
Atributs	
Efectivitat	<ul style="list-style-type: none"> <li>• Tasques com creació de partides, fer una jugada, enviar un missatge al xat... es poden fer ràpidament i és difícil equivocar-se en el procés.</li> <li>• Gairebé totes les funcions es poden completar correctament la primera vegada que s'utilitzen.</li> </ul>
Eficiència	<ul style="list-style-type: none"> <li>• El número d'accions necessàries per a completar una tasca és el mínim possible.</li> <li>• Un usuari novell pot adquirir l'agilitat d'ús d'un usuari expert en molt poc temps.</li> <li>• Les pantalles que no són de joc, estan optimitzades per a què l'usuari no hi perdi el temps (lobby, creació partida, informació de l'usuari...).</li> </ul>
Satisfacció	<ul style="list-style-type: none"> <li>• El nivell de dificultat d'utilització de l'aplicació és prou baix. Ja que moltes de les tasques es fan de manera intuïtiva.</li> <li>• Tant la interfície gràfica com l'experiència d'usuari han estat molt cuidades per a què l'aplicació agradi, visiblement i funcionalment.</li> </ul>
Facilitat d'aprenentatge	<ul style="list-style-type: none"> <li>• La corba d'aprenentatge no és molt elevada, ja que els conceptes d'aquest són senzills i clars.</li> <li>• L'aplicació és molt fàcil d'utilitzar i el seu aprenentatge complet és ràpid.</li> </ul>

Contingut	<ul style="list-style-type: none"> <li>• S'ha intentat que cap pantalla tingui una sobre informació exagerada per tal de no confondre a l'usuari.</li> <li>• No hi ha pàgines innecessàries.</li> </ul>
Portabilitat	<ul style="list-style-type: none"> <li>• El software només necessita connexió a internet per a funcionar.</li> </ul>
Context	<ul style="list-style-type: none"> <li>• És possible utilitzar l'aplicació desde qualsevol ubicació, sempre que disposem de connexió a internet.</li> <li>• El dispositiu ha de ser un smartphone o tauleta amb un sistema operatiu no obsolet.</li> </ul>

A part de tenir en compte tots aquests atributs a l'hora de desenvolupar l'aplicació, s'han estudiat i tingut molt en compte les diferents guies de usabilitat i principis de disseny oficials de *iOS* i *Android*.

Tot i això, en alguns casos ha estat difícil seguir aquestes guies, ja que no estem desenvolupant aplicacions natives per a cada sistema operatiu mòbil.

## 10.7 Disseny de la Base de Dades

Com s'ha explicat anteriorment, la Base de Dades que utilitzarem per a emmagatzemar totes les dades del sistema serà no relacional o NoSQL. Per tant, per a explicar el disseny d'aquesta, es mostraran les diferents estructures d'objectes JSON que es guardaran. Per entendre millor les dades, es faràn captures de l'eina de visualització de dades de Firebase.

### 10.7.1 Users

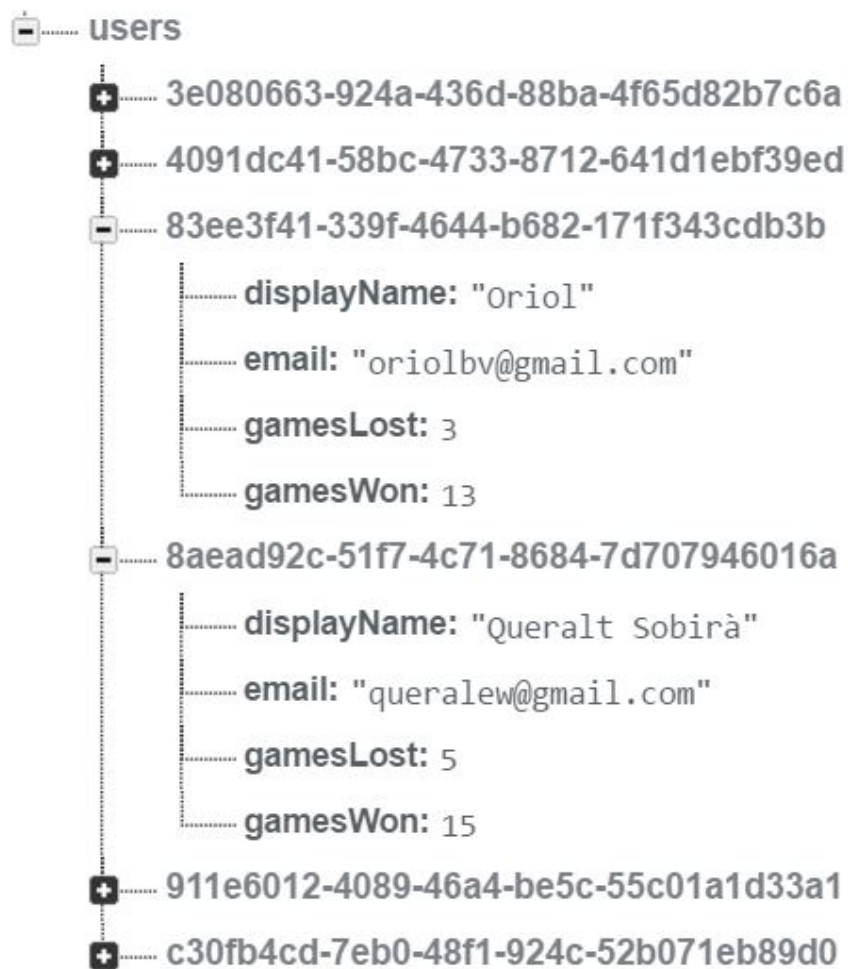


Figura 32: Esquema JSON dels usuaris del sistema

Com es pot veure a la figura anterior (Figura 32), de cada usuari en guardarem, el seu *nickname*.

A més a més, guardarem el número de partides guanyades i perdudes, per a omplir el gràfic d'informació d'usuari i el ranking global del joc.

### 10.7.2 Games

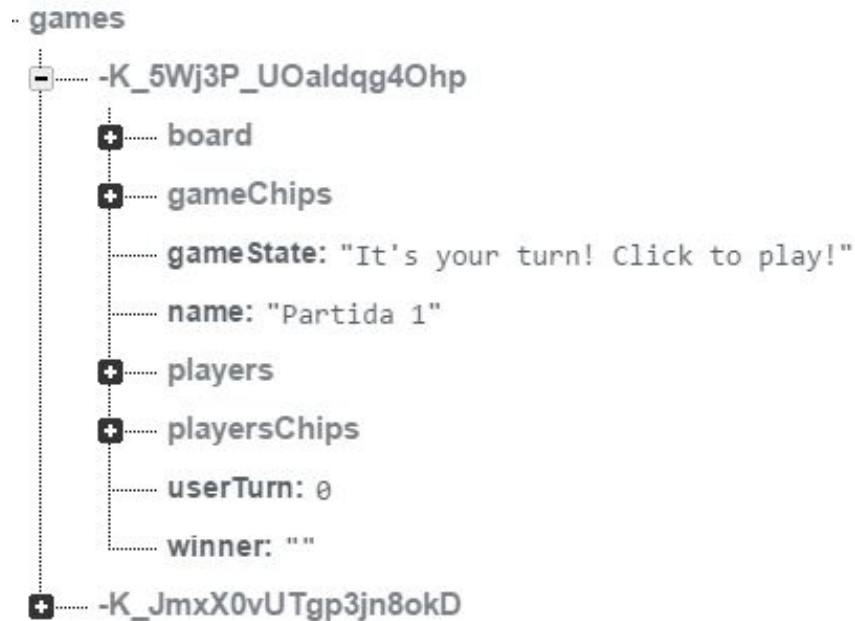


Figura 33: Esquema JSON de les partides

A l'objecte *games* hi guardarem totes les dades rellevants per a portar a terme una partida. Entre aquests, tindrem l'estat actual del taulell de la partida, amb totes les fitxes col·locades, les fitxes de la partida, tant les que tenen els jugadors d'aquesta, com les que s'utilitzaran per repartir durant el joc (quan un jugador passi el torn, per exemple).

També disposarem del camp *userTurn*, que ens indicarà l'índex del jugador que li toca tirar en aquest precís moment.

### 10.7.3 Chats

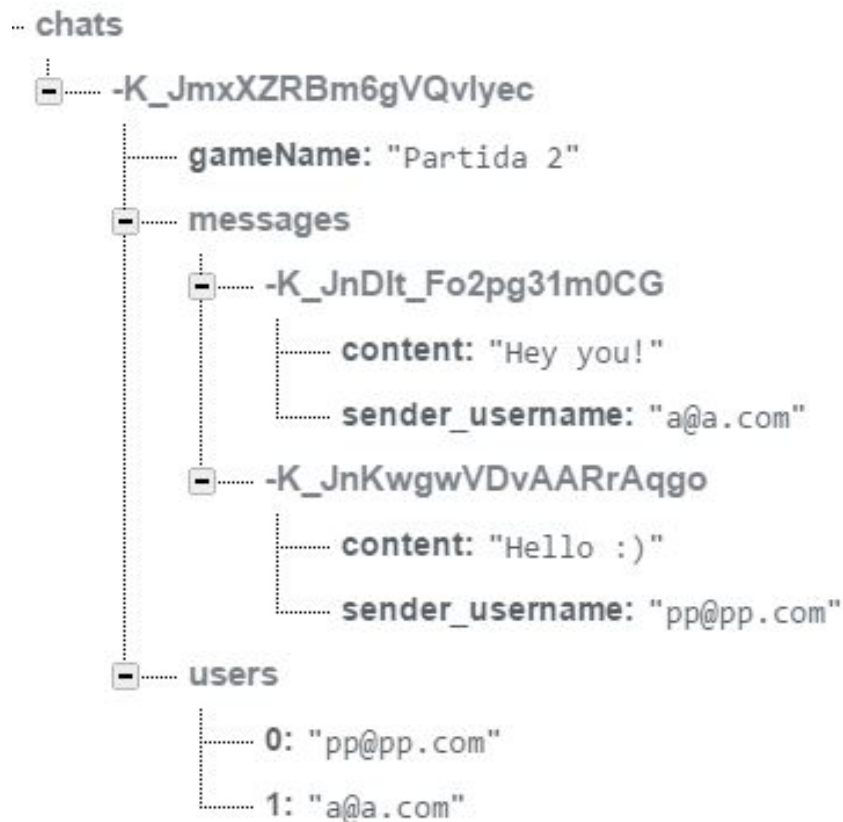


Figura 34: Esquema JSON dels chats actius en el sistema

En la figura anterior (Figura 34), es mostra un exemple d'un xat de partida guardat al nostre sistema.

En aquest objecte guardarem, el nom de la partida, els usuaris que hi participen i tots els missatges que s'han enviat en el xat.

Per a cada missatge del xat, guardarem el contingut d'aquest (*content*) i l'usuari que l'ha enviat (*sender\_username*).

# 11 Desenvolupament

## 11.1 Funcionalitats

Seguidament, es descriuràn les funcionalitats més rellevants de l'aplicació. Tot i això, no es mostrarà tot el codi, ja que estarà disponible al repositori de Github del projecte ("https://github.com/oriolbv/Scramikub").

### 11.1.1 Funcions d'administració d'usuaris

En el cas de les funcions de login i registre d'usuaris, utilitzarem els mètodes que ens aporta la llibreria de Firebase per a aplicacions desenvolupades en AngularJS, anomenada *angularfire*.

Per a que això funcioni correctament, necessitem activar la propietat de *Email and Password Authentication* al panell de control de Firebase.

#### Login

```
1 $scope.login = function(user) {
2   if (user && user.email && user.password) {
3     $ionicLoading.show({
4       template: 'Signing In...'
5     });
6     auth.$authWithPassword({
7       email: user.email,
8       password: user.password
9     }).then(function(authData) {
10      console.log("Logged in as:" + authData.uid);
11      $ionicLoading.hide();
12      userConnected = authData;
13      // Redirect to the lobby page
14      $state.go('initial');
15    }).catch(function(error) {
16      alert("Authentication failed:" + error.message);
17      $ionicLoading.hide();
18    });
19  } else {
20    alert("Please enter email and password both");
21  }
22 };
```

Listing 1: Login d'usuari

Com es pot observar en el codi anterior, s'utilitzarà el mètode *\$authWithPassword*. Aquest, enviarà l'email i la contrasenya de l'usuari, que ens arriben per la vista i comprovarà si aquest existeix al sistema i la contrasenya insertada és la

correcta.

En el cas de que la contrasenya sigui correcta, guardarem per a ús intern les dades del usuari, i serem redirigits a la pantalla del Lobby automàticament.

Per altra banda, si l'usuari i/o contrasenya són incorrectes, s'ens alertarà de que hi ha hagut un error en el procés de login i no podrem continuar endavant.

## Registre

Per a què un usuari es pugui registrar, una vegada es trobi a la pantalla de registre, haurà d'omplir en el formulari el seu nom (*Display Name*), email i contrasenya.

```
1 $scope.createUser = function (user) {
2   console.log("Create User Function called");
3   if (user && user.email && user.password && user.displayname) {
4     $ionicLoading.show({
5       template: 'Signing Up...'
6     });
7
8     auth.$createUser({
9       email: user.email,
10      password: user.password
11    }).then(function (userData) {
12      alert("User created successfully!");
13      ref.child("users").child(userData.uid).set({
14        email: user.email,
15        displayName: user.displayname
16      });
17      $ionicLoading.hide();
18      $scope.modal.hide();
19    }).catch(function (error) {
20      alert("Error: " + error);
21      $ionicLoading.hide();
22    });
23   } else
24     alert("Please fill all details");
25 };
```

Listing 2: Registre d'usuari

En el cas de que l'usuari no hagi omplert tots els camps, apareixerà un avís indicant-ho.

Per a crear un nou usuari a la Base de Dades amb un nou identificador únic (*uid*), s'utilitzarà el mètode *\$createUser*. Aquest mètode comprovarà que no existeixi ja un usuari amb el mateix email. En el cas de que no existeixi, crearà un objecte amb l'identificador únic. Seguidament, afegirem la informació que volguem en aquest objecte, en el nostre cas, l'email i el nom de l'usuari, però també podríem afegir



altres camps, com l'any de naixement, una foto de perfil...

Si el email ja existeix a la Base de Dades del sistema, apareixerà un avís indicant-ho.

El *modal*, guardat en el scope del controlador, és la plantilla html de la pantalla de registre (*signup.html*)

### 11.1.2 Funcions d'administració de partides

En aquest apartat es mostrarà la part del llistat de les partides actives en aquest mateix moment. En aquest cas, ens centrarem en el codi HTML per a veure com funcionen els components de Ionic.

#### Lobby

```
1 <ion-view>
2   <ion-content>
3     <button class="button button-clear button-positive" ng-click="addGame()">Create new game!</button>
4     <div>
5       <ion-list class="list list-inset" show-delete="shouldShowDelete" style="width:80px">
6         <div class="item item-divider">
7           Games
8         </div>
9         <ion-item ng-repeat="game in gamesToPlay track by $index" ng-click="clickGame(game)">
10          <h2>{{game.name}}</h2>
11          <h3>{{game.players[0]}} VS {{game.players[1]}}</h3>
12          <h3>{{game.gameState}}</h3>
13        </ion-item>
14      </ion-list>
15    </div>
16  </ion-content>
17 </ion-view>
```

Figura 35: Codi HTML del Lobby

Com es pot observar en el codi anterior, utilitzarem el component *ion-list* per a llistar totes les partides actives de l'usuari.

L'altre aspecte interessant són els *bindings* que utilitzarem per a poder llistar les dades del controlador a la vista. En aquest cas, fem un *binding* de l'array de partides (*gamesToPlay*).

Una altra característica interessant de **AngularJS** és la sentència *ng-repeat* que utilitzarem per a recórrer l'array de partides i inserir cadascuna d'aquestes en un item de la llista.

Ara es mostrarà com s'obtenen les partides actives de l'usuari des del controlador:

```
1 $scope.games.$loaded().then(function (games) {
2   var auxGames = [];
3   for (var i = 0; i < $scope.games.length; ++i) {
```

```

4     var game = $scope.games[i];
5     for (var j = 0; j < game.players.length; ++j) {
6         if (game.players[j] == userConnected.password.email) {
7             if (game.players[game.userTurn] == userConnected.
                password.email) {
8                 game.gameState = "It's your turn! Click to play!"
                "
9                 $scope.gamesToPlay.push(game);
10            }
11            else {
12                game.gameState = "It's not your turn. Waiting
                    for other users..."
13                $scope.gamesToPlay.push(game);
14            }
15            auxGames.push(game);
16        }
17    }
18 }
19 $scope.games = auxGames;
20 });

```

Listing 3: Get de les partides de l'usuari

Aquesta funció recorrerà totes les partides on hi participi l'usuari i comprovarà si li toca o no tirar. Això es farà per a mostrar un missatge o un altre a la vista.

### 11.1.3 Implementació del procés de joc

#### Taulell

La implementació de la vista del joc (on es troba la representació del taulell, les fitxes de joc...) ha sigut la part més complexa de desenvolupar, ja que conté molta interacció amb l'usuari i era primordial que l'experiència d'aquest fos molt bona.

```

1  <ion-scroll zooming="true" overflow-scroll="false" direction="xy" style = "height: 70%" delegate-handle="myScroll" scroll="false">
2    <div class = "scroll-container">
3      <table id="table-board" class="boardgame" >
4        <tr ng-repeat="(i,squares) in board" on-finish-render="ngRepeatFinished">
5          <td id="cell-drop" class="square" ng-repeat="(j,square) in squares track by $index"
6            ng-drop="true" ng-drop-success="onDropComplete($data,$event,i,j)"
7            ng-click="changeChipPosition($data,$event,i,j)"></td>
8        </tr>
9      </table>
10    </div>
11  </ion-scroll>

```

Figura 36: Codi HTML del Taulell

En el codi anterior, podem observar que el taulell es troba en un element capaç de fer *scroll* i *zoom*, fent així que l'usuari es pugui moure per tot el taulell i augmentar la visibilitat quan li sembli.

El taulell és una taula, on cada casella d'aquesta és una posició on hi podem posar qualsevol fitxa. En aquesta part del codi també podem observar els events que utilitzarem:

- **ng-drop-success** : Aquest event s'executarà quan deixem anar una fitxa sobre una posició del taulell, finalitzant així l'arrossegament de la fitxa del jugador (**Drag and Drop**).
- **ng-click** : En el cas de que volguem canviar de posició una fitxa que ja es troba en el taulell, tocarem la fitxa en qüestió i, seguidament, tocarem la posició on la volem moure.  
En aquest procés l'event *ng-click* s'executarà dos vegades, una per escollir la fitxa i l'altra per recol·locar-la.

A la part posterior de la pantalla es trobaran totes les fitxes que té el jugador en aquell precís moment.

```
1 <div>
2   <div style="background: rgba(78, 0, 0, 0.7);" id="chips-div" class="section div-chips">
3     <div ng-repeat="obj in items">
4       <div class="square-chip" ng-drag="true" ng-drag-data="obj"
5         data-allow-transform="false" on-drag="onDrag()" ng-drag-scroll>
6         
7       </div>
8     </div>
9   </div>
10 </div>
11
12 <div ng-class="{disabled:pager.currentPage === 1}" style="float: left; width: 10%;">
13   <button class="button button-clear button-positive" ng-click="setPage(pager.currentPage - 1)">
14     Previous
15   </button>
16 </div>
17 <div ng-class="{disabled:pager.currentPage === pager.totalPages}" style="float: right; width: 10%;">
18   <button class="button button-clear button-positive" ng-click="setPage(pager.currentPage + 1)">
19     Next
20   </button>
21 </div>
```

Figura 37: Codi HTML de les fitxes del jugador

Primer de tot, tindrem un element que contindrà totes les fitxes que caben a la pantalla del nostre dispositiu. Aquestes estaran ordenades en dos files de com a mínim 5 fitxes cadascuna.

Totes les fitxes podran ser arrossegades cap al taulell (**Drag**).

Seguidament tindrem dos botons inferiors que ens permetran, mitjançant un sistema de paginació, veure les fitxes que actualment no s'estaven mostrant a la pantalla.

## Comprovacions taulell

En el moment en que l'usuari prem el botó de *Play*, el sistema comprovarà si el taulell actual és correcte o no. En el cas de que sigui correcte, la jugada serà tramesa. En cas contrari, no es deixarà que l'usuari acabi la seva jugada, per tant, haurà de modificar-la, o bé, passar el seu torn i ser penalitzat amb una fitxa aleatòria.

La funció encarregada de validar l'estat del taulell és la següent:

```
1 function checkBoardGame(boardgame) { ... }
```

Listing 4: Capçalera funció de comprovacions del taulell

En aquesta funció li passarem per paràmetre una matriu que representa el taulell actual, i aquesta ens retornarà un booleà depenent de si el taulell és correcte o no.

```
1 for (var i = 0; i < verticalSize; ++i) {
2   for (var j = 0; j < horizontalSize; ++j) {
3     // If there is a chip in actual position...
4     if (boardgame[i][j].value != 0) {
5       var actualCell = {
6         "color" : boardgame[i][j].color,
7         "value" : boardgame[i][j].value
8       };
9       ...
10    }
11  }
12 }
```

Listing 5: Capçalera funció de comprovacions del taulell

El que farà la funció, serà recórrer tota la matriu *boardgame* buscant posicions que no estiguin buides, és a dir, que hi hagi una fitxa.

Una vegada es trobi una posició del taulell amb una fitxa, es guardarà el color i valor d'aquesta i es començarà a recórrer les posicions de la fila i column on ens trobem, per comprovar si aquesta fitxa és part d'una combinació o no.

```
1 // Check all the row positions
2 var rowSet = [actualCell];
3 // ---->
4 for (var rowRight = j+1; rowRight < verticalSize; ++rowRight) {
5   if (rowRight != j) {
6     if (boardgame[i][rowRight].value == 0) { break; }
7     else {
8       var rightCell = {
9         "color" : boardgame[i][rowRight].color,
10        "value" : boardgame[i][rowRight].value
11      };
12    }
```

```

13     var previousCell = rowSet[rowSet.length - 1];
14     // STRAIGHT OF SAME COLOR
15     if ((previousCell.color == rightCell.color) || (
16         rightCell.color == "joker")) {
17         if (previousCell.value != 15) {
18             if ((rightCell.value == previousCell.value +
19                 1) || (rightCell.value == previousCell.
20                     value - 1) || (rightCell.color == "joker"))
21                 {
22                     rowSet.push(rightCell);
23                 }
24             } else {
25                 if ((rightCell.value == 1) || (rightCell.color
26                     == "joker")) {
27                     rowSet.push(rightCell);
28                 }
29             }
30         }
31     }
32 }
33
34 // <-----
35 . . .

```

En el fragment de codi anterior, es pot observar com es fa un recorregut de la fila on anteriorment ens hem trobat una fitxa.

En aquest cas, recorrerem la fila d'esquerra a dreta començant per la següent a la posició actual.

Primer de tot, farem sempre la comprovació de que la fitxa següent (la de la nostra dreta), no és la fitxa actual, ja que, com que el taulell és cíclic (una fitxa que es troba a un costat del taulell pot combinar amb la del costat oposat), podríem entrar en un bucle infinit.

Seguidament comprovarem quin tipus de combinació s'està fent en aquesta fila, una escala del mateix color, o bé, una combinació de colors amb el mateix valor.

En el cas de que la fitxa de la dreta pugui formar una combinació, l'afegirem en un *array* que posteriorment analitzarem.

```

1 if (isSetCorrect(rowSet)) {
2     if (!containsSet(boardSets, rowSet)) {
3         boardSets.push(rowSet);
4     }

```

```

5 } else {
6     if (rowSet.length > 1) {
7         isCorrect = false;
8         var sIncHorit = "";
9         for (var iHor = 0; iHor < rowSet.length; ++iHor) {
10             sIncHorit += rowSet[iHor].value + " " + rowSet[iHor].
                color + ", "
11         }
12         console.log("Incorrect Horizontal Set => [ " + sIncHorit +
            " ]");
13     }
14 }

```

Una vegada recorreguda la fila de la posició actual, comprovarem si la combinació que hem anat afegint a l'array *rowSet* està formada d'almenys 3 fitxes, i per tant és correcta.

En el cas de que sigui correcta, afegirem la combinació a l'array de combinacions correctes del taulell, en cas contrari, el taulell ja no serà correcte, ja que, no hi pot haver absolutament cap combinació incorrecte sobre el taulell.

La resta de codi d'aquesta part, només ens ajuda a millorar el codi escrivint per la consola del navegador les combinacions incorrectes que hi han en aquest moment.

Seguidament, faltirà comprovar que no hi hagi una combinació formada verticalment, per això faltirà repetir el procés anterior, però recorrent la columna de dalt a baix, i de baix a dalt.

Una vegada fet això, haurem comprovat si la fitxa de la posició actual forma una combinació correcta o no.

Al final del bucle, haurem recorregut tot el taulell i sabrem si hi ha alguna combinació incorrecta, o bé el taulell és vàlid per a què l'usuari acabi la seva jugada.

#### 11.1.4 Altres funcionalitats

##### Xat

Els diferents xats que tinguem actius en el sistema, seràn guardats amb un únic identificador, que utilitzarem per a obtenir i afegir missatges del xat de la partida corresponent.

Seguidament, es mostra la funció per a enviar un missatge a tots els usuaris de la partida en qüestió.

```
1 $scope.sendMessage = function(newMessage) {
2     var refChat = new Firebase(firebaseUrl + "/chats/" + $scope.
      chat.$id + "/messages");
3     $firebaseArray(refChat).$add({
4         "content" : newMessage,
5         "sender_username" : userConnected.password.email
6     });
7     for (var i = 0; i < $scope.chats.length; ++i) {
8         if ($scope.chats[i].gameName == $scope.actualGame.name) {
9             $scope.chat = $scope.chats[i];
10        }
11    }
12    $scope.messages = $scope.chat.messages;
13    $ionicScrollDelegate.$getByHandle('chatScroll').scrollBottom(
      true);
14    $scope.newMessage = "";
15 }
```

Listing 6: Enviar missatge al xat d'una partida

Com es pot observar, obtenim l'objecte amb l'identificador únic del chat corresponent. Després afegirem un fill a aquest objecte, amb el contingut del missatge i l'email de l'usuari que l'ha enviat.

##### Ranking

En aquesta pantalla es llistaràn els 5 millors jugadors de Scramikub del moment. Això es calcularà amb la diferència de partides guanyades i perdudes de cada usuari. D'aquesta manera el ranking serà just per a tots els jugadors.

##### Informació d'usuari

Per a llistar tota l'informació de l'usuari connectat, simplement obtindrem la seva informació de la nostra Base de Dades, on també s'hi troben les partides guanyades i perdudes d'aquest usuari en concret.

Per a representar les partides guanyades i perdudes de l'usuari, utilitzarem un component de *Chart.js*. Aquesta llibreria ens permetrà inserir un gràfic fàcilment

i que només ens haguem de preocupar de les dades que aquest mostrarà.  
En el meu cas, s'ha utilitzat un gràfic de tipus *pie*:

```
1 <div class="item item-text-wrap div-chart">
2   <canvas id="pie" class="chart chart-pie chart-win-lose"
3     chart-data="data" chart-labels="labels" chart-options="options">
4   </canvas>
5 </div>
```

Figura 38: Codi HTML del gràfic d'usuari

Una vegada tinguem el codi de la interfície només ens caldrà fer el *binding* de les dades que es troben en el controlador de la vista d'informació d'usuari.

```
1 $scope.data = [userInfo.gamesWon, userInfo.gamesLost];
```

Listing 7: Binding de les dades del gràfic

Com es pot veure, a la col·lecció *data* li afegirem el número de partides guanyades i perdudes de l'usuari.

### 11.1.5 Administrador extern de l'aplicació

Per tal d'administrar el sistema externament i tenir un control total sobre aquest, s'ha desenvolupat una aplicació web simple.

Aquesta web serà un simple **CRUD** (*Create*, *Read*, *Update* i *Delete*) de les dades del sistema, a més a més d'una part d'analítiques on podrem veure com està funcionant l'aplicació mòbil.

Cal dir que la web serà totalment responsive, per tal de que el futur administrador del joc pugui utilitzar-la desde del navegador del seu mòbil o tauleta.

Durant el transcurs d'aquest projecte, s'ha utilitzat aquesta plataforma desde un localhost, i no s'ha desplegat enlloc.

### Usuaris

A la pantalla d'usuaris veurem un llistat d'usuaris amb el seu nom i email. Podrem editar i esborrar els usuaris existents en el sistema, o bé crear usuaris nous amb un formulari similar al de l'aplicació mòbil.

En les següents figures es pot veure el resultat final d'aquesta pàgina:



Games	Users	Analytics
-------	-------	-----------

# Users

New

Name	Email	Actions
Enric Verdaguer	enric123@gmail.com	<div>Edit</div> <div>Eliminar</div>
gas	gas@gas.com	<div>Edit</div> <div>Eliminar</div>
pp	pp@pp.com	<div>Edit</div> <div>Eliminar</div>
Queralt Sobirà	queralew@gmail.com	<div>Edit</div> <div>Eliminar</div>
a	a@a.com	<div>Edit</div> <div>Eliminar</div>
Oriol_Burgaya	oriol@gmail.com	<div>Edit</div> <div>Eliminar</div>
ss	ss@hotmail.com	<div>Edit</div> <div>Eliminar</div>

Figura 39: Pantalla d'administració d'Usuaris

Users

Analytics

rs

New

daguer

obirà

a@a.com

Edit

Eliminar

New User

Display Name

Email

Password

Add

Cancel

Figura 40: Pantalla de creació d'usuaris

## Partides

En el cas de les partides, s'ens llistaràn totes les partides actives en el sistema, podrem editar el seu nom i també eliminar-les però, en aquest cas, només serà possible crear partides desde l'aplicació mòbil.

Games	Users	Analytics
-------	-------	-----------

Games		
Name	UserTurn	Actions
Partida 1	a@a.com	<a href="#">Edit</a> <a href="#">Eliminar</a>
Partida 2	queralew@gmail.com	<a href="#">Edit</a> <a href="#">Eliminar</a>
Partida 3	oriol@gmail.com	<a href="#">Edit</a> <a href="#">Eliminar</a>

Figura 41: Pantalla d'administració de Partides

## Analítiques

En aquesta pantalla apareixeràn diverses gràfiques molt explicatives que representen algunes dades interessants del sistema:

- **Usuaris creats:** Aquesta gràfica ens indicarà quants usuaris han estat creats en el sistema cada mes de l'any. En aquest moment aquesta gràfica no és del tot fiable, ja que els mesos representats han sigut de desenvolupament i s'han creat molts usuaris per a fer testing d'algunes de les features.
- **Partides creades:** En aquesta gràfica veurem les partides creades en el sistema en cada mes de l'any. Ara mateix tampoc tenim dades rellevants, ja que els mesos representats han sigut de desenvolupament.
- **Ranking:** En aquest Ranking, podrem veure la diferència de partides guanyades i perdudes de tots els jugadors registrats en el sistema, per tant, ens donarà més visibilitat que no pas el ranking que veuen els usuaris a l'aplicació.

Active games : 3

### Created users :



### Created Games :



Figura 42: Gràfiques d'usuaris i partides creades

## 12 Adaptacions realitzades

En aquest apartat es llistaràn totes les adaptacions fetes al joc per tal de que l'aplicació resultant sigui el més *user-friendly* possible.

### 12.1 Adaptacions del reglament

- **Mecànica a l'hora de passar el torn:** En el joc original de taula, una vegada el jugador no pot fer cap moviment amb cap de les seves fitxes, aquest té la capacitat d'agafar una fitxa del 'sac' i intentar jugar amb aquesta fitxa. Per tal de no complicar la mecànica de joc de l'aplicació mòbil, aquesta funció ha estat modificada de manera que, si el jugador no pot tirar, aquest haurà de passar el torn i se li repartirà un fitxa aleatòria amb la qual no podrà jugar aquell mateix torn.
- **Temps limitat per torn:** En la versió inicial de l'aplicació s'ha eliminat la opció d'utilitzar un cronòmetre per a limitar el temps del torn del jugador. Això s'ha fet per a què els usuaris no experimentats puguin prendre's el seu temps a l'hora de fer una jugada. Tot i això, no es descarta que en una versió futura de l'aplicació es doni la possibilitat de limitar el temps de les partides, però mai obligatòriament.

### 12.2 Adaptacions gràfiques

- **Forma de les fitxes:** A diferència del joc original, les fitxes de l'adaptació a dispositius mòbils seràn de forma rectangular i no circular. Això es va decidir després d'estudiar les diferents aplicacions d'interfície gràfica similar. També es va veure que la superfície per a interactuar amb la fitxa era més gran en el cas que aquesta fós rectangular.

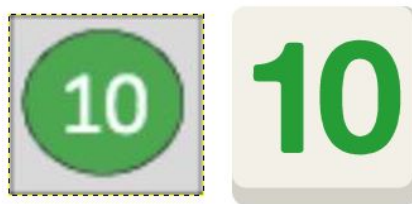


Figura 43: Diferència fitxes

- **Forma del taulell:** A causa del canvi de forma de les fitxes del joc, el taulell de joc també va canviar per a què donés la sensació de que aquestes encaixaven en qualsevol posició.

## 13 Problemes trobats durant el desenvolupament

En aquest apartat es recolliran els obstacles més importants que m'he anat trobant durant el desenvolupament d'aquest projecte.

### 13.1 CrossPlatform

Com s'ha especificat anteriorment, un dels requisits del sistema que s'ha desenvolupat és que aquest sigui multiplataforma o *cross-platform*.

Tot i que les eines de desenvolupament ens simplifiquen en gran part la vida a l'hora de compilar l'aplicació en els diferents sistemes operatius mòbils, el comportament d'aquesta pot variar completament en cada un d'ells.

Això m'ho he trobat, sobretot, amb alguns controls HTML, que tenien un comportament totalment diferent en cada sistema operatiu.

Un dels punts més crítics de l'aplicació on em vaig trobar amb aquest problema, va ser en la funcionalitat de *scroll* i *zoom* del taulell. Per a solucionar-ho vaig haver d'invertir una bona part del meu temps en provar diferents maneres d'implementar aquestes funcionalitats fins a trobar-ne una que funcionés correctament en els dos sistemes operatius.

### 13.2 Native gestures

Per a emular els gestos nadius que es fan a Android i iOS, Ionic utilitza diverses directives d'AngularJS i Javascript pur, com *ng-click*, *touchstart*...

El problema me'l vaig trobar a l'hora de desenvolupar el *drag and drop* de les fitxes cap al taulell, ja que el comportament de la directiva *ng-drag* (la més comuna per a fer aquesta tasca), era diferent en cada sistema operatiu.

El següent problema que em vaig trobar amb aquest gest natiu, va ser la impossibilitat d'utilitzar-lo quan l'element que volíem arrossegar es trobava dins d'un element *scrollable*. Això em va fer canviar la manera de canviar de posició les fitxa que ja es troben en el taulell, fent click a la fitxa que volem canviar de posició i, seguidament un altre click a la nova posició on la volem moure.

Tot i això, el resultat final és bastant bo i, el que és més important, és molt fàcil d'utilitzar.

### 13.3 Rendiment

Tot i el ràpid desenvolupament que comporta l'ús d'eines web, si aquest no s'ha fet amb molta cura, pot comportar també molts problemes en el rendiment de l'aplicació una vegada compilada per als sistemes operatius mòbils corresponents. En el cas de l'aplicació que s'ha tractat en aquest projecte, la part on hi ha hagut

més problemes de rendiment ha estat en la pantalla del taulell de joc, ja que es carreguen moltes imatges (de fitxes i background) al mateix moment, i això pot provocar un cert alentiment en el sistema.

Això em va obligar a modificar la forma amb la que carregava el background del taulell per tal de millorar el rendiment al dirigir-nos a aquesta pantalla.

## 14 Conclusions i Treball Futur

Una vegada acabat el període del projecte, he obtingut varies conclusions sobre el resultat obtingut d'aquest.

En primer lloc, cal remarcar que s'ha assolit clarament l'objectiu d'aquest projecte, ja que, s'ha desenvolupat amb èxit una aplicació mòbil cross-platform i online que adapta el joc de taula Scramikub.

Seguidament, m'agradaria remarcar també les dificultats i obstacles que s'han trobat sobretot durant el desenvolupament d'aquest projecte, ja que, al ser un joc, s'allunya de totes les aplicacions més estàndard, i per tant no es pot trobar gairebé cap exemple per internet.

L'adaptació del joc a una aplicació mòbil, ha ajudat per a plantejar-se diverses coses sobre el joc de taula original: veure quines regles de joc són més interessants i quines podrien canviar-se, quins aspectes del joc poden ser més complicats per usuaris novells...

En quan a la gestió del projecte, penso que la metodologia utilitzada (SCRUM) ha agilitzat moltíssim el procés de desenvolupament de l'aplicació i ha permès tenir un cert control sobre les hores invertides i les desviacions temporals que s'han donat.

Gràcies a l'eina de gestió de projectes (JIRA), ha estat molt més fàcil gestionar el desenvolupament de l'aplicació.

Finalment, m'agradaria valorar positivament el resultat d'aquest projecte, on s'han estudiat tecnologies molt noves i vàlides per a llançar una aplicació mòbil al mercat, donant així una alternativa al desenvolupament d'aplicacions natives.

Durant tota l'execució d'aquest projecte, s'han complert totes les competències tècniques que es van definir a l'inici.

De totes aquestes competències, m'agradaria remarcar la importància que ha tingut la de tenir un control de la qualitat del producte, ja que, sense aquest control durant la fase de desenvolupament, segurament no s'haurien obtingut uns resultats tant positius.

Tot i això, no hi ha cap competència tècnica que hagi quedat apartada durant el projecte.

Com a treball futur, hi haurien moltes millores interessants per a implementar en aquesta aplicació, com per exemple, millorar la part d'informació d'usuari, o bé, afegir un mode per a jugar contra l'aplicació. També seria interessant la integració del joc a la plataforma Windows Phone, que tot i no ser molt popular, té un volum d'usuaris considerable.



## Referències

- [1] Android material usability guidelines. <https://material.io/guidelines/usability/accessibility.html>. (Consulta 2 Novembre 2016).
- [2] Angular js. <https://angularjs.org/>. (Consulta 23 Octubre 2016).
- [3] Apple, ipad air. <http://www.apple.com/shop/buy-ipad/ipad-air-2>. (Consulta 10 octubre 2016).
- [4] Apple, macbook air. <http://www.apple.com/shop/buy-mac/macbook-air>. (Consulta 10 octubre 2016).
- [5] Celoxis. <https://app.celoxis.com/>. (Consulta 3 Octubre 2016).
- [6] Design principles ios. <https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/>. (Consulta 2 Novembre 2016).
- [7] Fib - assignatures del grau. <http://www.fib.upc.edu/fib/estudiar-enginyeria-informatica/assignatures.html>. (Consulta 23 Octubre 2016).
- [8] Firebase. <https://www.firebase.com/docs/>. (Consulta 26 Setembre 2016).
- [9] Gantter. <https://www.smartapp.com/>. (Consulta 3 Octubre 2016).
- [10] Hasbro. <http://www.hasbro.com/en-us/brands/hasbrogames>. (Consulta 26 Setembre 2016).
- [11] Ionic framework. <http://ionicframework.com/docs/>. (Consulta 26 Setembre 2016).
- [12] Jira, atlassian software. <https://www.atlassian.com/software/jira/features>. (Consulta 26 Setembre 2016).
- [13] One plus x. <https://oneplus.net/es/x>. (Consulta 10 octubre 2016).
- [14] Page personnel. <http://www.pagepersonnel.es/prensa-estudios/estudios-remuneracion>. (Consulta 10 octubre 2016).
- [15] Spryfox. <http://spryfox.com/our-games/alphabear/>. (Consulta 26 Setembre 2016).
- [16] Yushino. <http://www.yushino.com/>. (Consulta 26 Setembre 2016).

- [17] Zynga. <https://www.zynga.com/games/words-friends>. (Consulta 26 Setembre 2016).
- [18] Autor Desconegut. Plantilla de especificación de requisitos de software. (Consulta 2 Novembre 2016).
- [19] Enriquez Juan Gabriel. Usabilidad en aplicaciones móviles. (Consulta 5 Gener 2017).
- [20] Arkaitz Garro. *Advanced HTML5*. 2014. (Consulta 24 Novembre 2016).
- [21] UPC. Apunts arquitectura del software. 2015. (Consulta 12 Gener 2017).
- [22] UPC. Apunts arquitectura del software web. 2015. (Consulta 5 Gener 2017).